AD-A035 290 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 6/4

THE SCALING PROBLEM IN VISUAL PATTERN RECOGNITION.(U)

DEC 76 D D CARPENTER

UNCLASSIFIED GE/EE/76D-17 NL

1 OF 2

AD

A035290

GE/EE/76D-17

1

THE SCALING PROBLEM IN VISUAL PATTERN RECOGNITION

THESIS

GE/EE/76D-17

Douglas D. Carpenter
Captain USAF

14

GE/EE/76D-17

6

THE SCALING PROBLEM IN VISUAL PATTERN RECOGNITION.

9 Master's thesis,

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

10 Douglas D. Carpenter, B.S. 12 122p.

Captain USAF

Graduate Electrical Engineering

11 Dec[redacted]976

Approved for public release; distribution unlimited

012225

LB

Preface

The investigations described in this thesis are based on a model of the human visual process developed by Dr. Matthew Kabrisky. The research was directed toward extending the model to adequately account for the human ability to carry out "scale invariant pattern recognition". The project provided an opportunity to design and build a computer-based testing system, and to become acquainted with the complex problems of developing human psychological experiments.

I am indebted to Dr. Kabrisky for his initial suggestion for the research topic, as well as his continued guidance. I wish to thank Dr. Larry Goble for his assistance in developing the experimental paradigms used to test the effect of scale change. I also wish to express my appreciation to my wife for her invaluable assistance and support.

Contents

Contents

List of Figures

List of Tables

Abstract

The human pattern recognition system has been modeled as a system utilizing a low-pass, spatially-filtered, Fourier transform to represent input patterns and stored prototypes. This model is used to infer the existence of possible mechanisms underlying the scale invariant aspect of the human pattern recognition system. Two hypotheses are suggested: scaling (size normalization) of input patterns, or the storage of multiple-prototypes based on size. Experiments are carried out to distinguish between the two mechanisms. It is concluded that both mechanisms are utilized at different levels within the visual process. A revised model is developed to adequately account for the experimental results.

THE SCALING PROBLEM IN VISUAL PATTERN RECOGNITION

I. Introduction

The human visual system incorporates an extremely effective pattern recognition system. In fact, the ability of the human visual system far surpasses the capabilities of any pattern recognition system yet built. The success of the human system has generated a great deal of effort among pattern recognition specialists as well as psychologists toward understanding the human visual processes. The objective of this effort has been to develop a model of the human visual system that can be used as the basis for the development of more effective pattern recognition systems.

Successful models of the visual system must not only functionally describe the system; they must reflect the known details of the psychological and physiological aspects of the visual processes. Models developed in the face of relative ignorance, as are models of the human visual system, must be provided with built-in flexibility. That is, the details of the model should be modifiable as required to meet additional requirements imposed by an increasing understanding of the actual system. However, the fundamental concepts, which serve as the core of a successful model, should remain unchanged.

One model, which meets these requirements, was suggested by Kabrisky in 1966 (Ref 17). Kabrisky examined known anatomical and physiological data in order to determine the nature of the mathematical functions the cortical structure could support. The model is based on the theory that the cortical connections of the brain could support Fourier or other similar transform computations. The essential feature of the model is that information processing within the visual system is carried out in the transform domain rather than in the image domain. Applications and extensions of the Kabrisky model proposed by Radoy (Ref 22) and Tallman (Refs 27 and 28) have resulted in a pattern recognition model based on an adaptive low-pass spatially filtered Fourier transform. Mahaffey developed another transform based on his analysis of cortical structure that was also found to classify patterns much the same as humans would (Ref 18). Carl showed that the Walsh transform produced equivalent results (Ref 4).

Algorithms derived from the Kabrisky model have been extensively tested in a variety of pattern recognition tasks. The technique has been applied to the analysis of such diverse patterns as Chinese characters (Ref 1), handprinted numeric characters (Refs 27 and 28), a wide range of aperiodic signals (Ref 13) and speech spectagrams (Ref 20). The successes merely show that the model

is useful, not that it is a conceptually valid representation of the human system.

The question of the conceptual validity of the model may be explored in terms of psychological correlates - how well does it explain or predict human behavior? Several studies have been carried out in these areas. Maher, in an initial psychological investigation of the Kabrisky model, correlated the output of a computer simulation of the model with human responses for ranked similarity of animal forms (Ref 19). Ginsburg, using blurred letters, showed that the low frequency spatial content provides the necessary information for humans to identify letters (Ref 9). In the same paper, Ginsburg showed that human identification errors of rotated letters were correlated with variance of Euclidean distance measures of the low-pass spatially filtered transform of the rotated letters.

The most interesting result of the search for psychological correlates has been in the application of the model to the analysis of optical illusions. Ginsburg and Ragsdale have both done work in this area (Refs 10 and 23). They showed that the removal of higher spatial frequency components by filtering in the transform domain would account for the effects produced by many of the classical illusions. This explanation has the advantage of being general, rather than having to rely on specific

geometric features or higher level cognitive processes applicable to each illusion.

The past success of the search for psychological correlates of the model logically suggests that further studies be undertaken. One area yet to be explored is the effect of image scale changes on the human pattern recognition process. Conclusions may be drawn from the model concerning the mechanisms required for scale invariant pattern recognition. This paper will demonstrate that human psychological data does, in fact, provide evidence that such mechanisms are incorporated into the human pattern recognition system.

The approach taken in this paper will be to first describe the general aspects of the Kabrisky model. Special emphasis will be placed on the effect of scale changes. A detailed description of psychological evidence for scaling in the human system will then be presented. Specific experiments carried out to test hypotheses derived from the model will be described. Finally, an extension of the model incorporating a scaling process supported by the psychological evidence will be proposed.

II. The Filtered Transform Model

Introduction

This chapter will describe the filtered transform model of the human visual system. A brief outline of the psysiological aspects of the human visual system on which the model is based will be presented first. The effects of image scale changes on the pattern recognition process will be examined in detail. Inferences will be drawn from the model concerning the mechanisms by which the human pattern recognition system may account for scale change.

Human Visual System

Figure 1 schematically illustrates the visual pathway from the two retinae back to the visual cortex. After the neural impulses leave the retinae they pass backward through the optic nerves. At the optic chiasm all the nerves from the nasal halves of the retinae cross to the opposite side where they are combined with the fibers from the opposite temporal retinae to form the optic tracts. The fibers of each optic tract synapse in the lateral geniculate body. From this relay point the optic radiations fan out to carry the impulses to the visual cortex in the calcarine area of the occipital lobe.

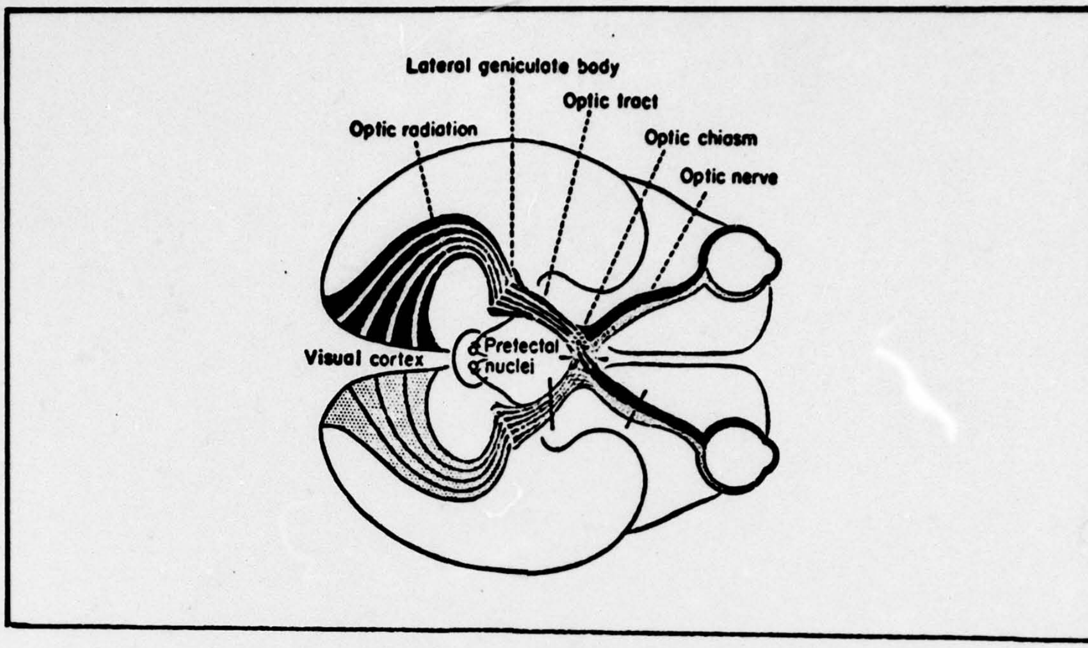

Fig. 1. The Visual Pathway (From Ref 12:626)

The visual pathway begins at the retina. The retina is a dense array of light sensitive receptors. Each retina contains about 125 million rods and about 5.5 million cones. These receptors are not uniformally distributed over the retina. Cone density increases as the optical axis is approached. In addition, fewer rods and cones converge on each optic nerve fiber, and the rods and cones become progressively more slender. Within the fovea, a very small area corresponding to the central two degrees of the visual field, there are no rods at all. Also, within this small area there appears to be a one to one correspondence between receptors, retinal ganglion cells and optic nerve fibers. This implies that there is little or no combining

at the retinal level of foval receptor output. This is in contrast to the periphery, where as many as 200 rods may share one retinal ganglion cell. These factors explain the high degree of visual acuity in the central portion of the retina in comparison with the very poor acuity in the peripheral portions (Ref 12:628).

The retinae project to the lateral geniculate bodies via the optic tract in a systematic topological manner. While the anatomy of the lateral geniculate bodies is well understood, their role in the visual process is not. Each lateral geniculate body is composed of six layers. Layers 2, 3, and 5 (from the surface inward) receive signals from the temporal portion of the ipsilateral retina, while layers 1, 4, and 6 receive signals from the nasal retina of the opposite eye. Because corresponding retinal fields in the two eyes connect with respective neurons that are approximately superimposed over each other in successive layers, it has been suggested that the lateral geniculate bodies play a major role in fusion of vision (Ref 12:629).

All layers of the lateral geniculate body relay visual information to the visual cortex. The boundaries between excited and nonexcited areas of the visual signals are considerably sharper in the lateral geniculate neurons than in the ganglion cells of the retina. This indicates that lateral inhibition probably occurs in the lateral geniculate body. This lateral inhibition enhances the degree of

contrast in the visual pattern.

The optic radiations project the visual information from the lateral geniculate to the primary visual cortex (Broadman's Area 17). The mapping of the information on Area 17 is homeomorphic to the retinal image. That is, adjacent receptive fields of the retina project to adjacent regions of the cortex.

The cerebral cortex is structurally a flat, thin sheet made up of six identifiable layers. It has been shown experimentally (Refs 15 and 16) that functionally the cortex consists of small columnar elements extending perpendicular to the surface. Kabrisky calls these small columns "basic computational elements" (Ref 17:39). It is known that there is a rich interconnection between columns of neighboring areas of the cortex; where the preponderence of connections are perpendicular to the surface, connections within the sheet are generally quite short (Ref 5).

The details of the interconnections between the bce's is unknown. However, the nature of the interconnectivity can be deduced from strichnine experiments such as those carried out by Dusser de Barenne and McCulloch (Ref 6). When they placed a small amount of strichnine on the surface of Area 17, they found that strichnine-derived pulses appeared at many widely separated isolated points in the surrounding visual association cortex. This seemingly

random interconnection of elements suggested to Kabrisky the possibility that this portion of the cortex was capable of carrying out mathmatical transforms. This initial assumption led to the development of the spatially filtered transform model of the visual process.

The Kabrisky Model

Kabrisky's analysis of the biological evidence led him to conceptualize the cortical areas 17, 18, and 19 as densely connected two-dimensional sheets as shown in Fig. 2.

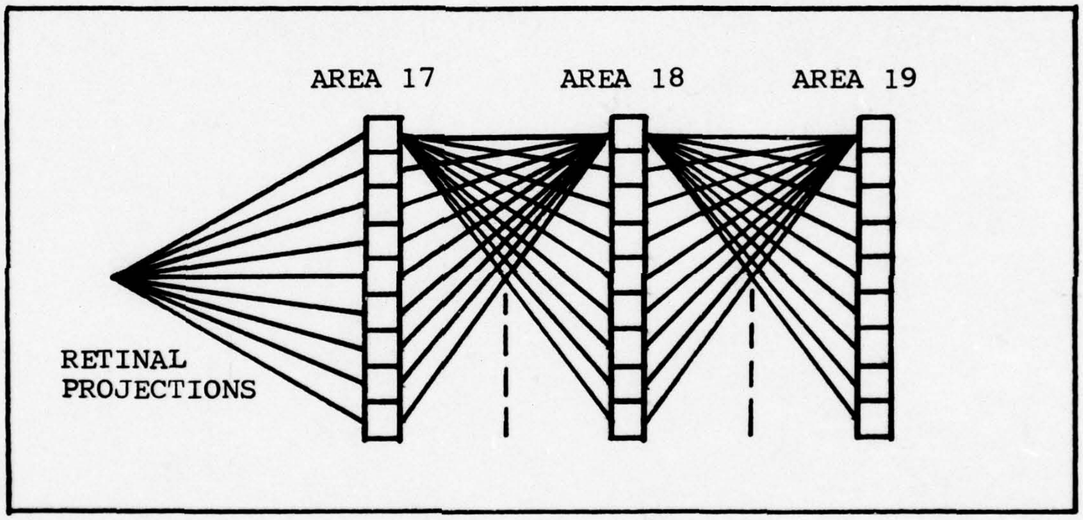

Fig. 2. Schematic Representation of Cortical Connections

His analysis of this structure suggested that it was capable of carrying out a cross-correlation between areas 17 and 18. He suggested that the pattern recognition might be based on this type of calculation. However, cross-correlation is equivalent to template matching, and as such,

it has several inherent disadvantages. Trivial changes to the input pattern can fault the system. Additionally, the technique would require large amounts of storage to maintain the stored templates. To overcome both of these problems, Kabrisky suggested that a low pass filtered two-dimensional Fourier Transform might be the actual cortical transform (Ref 17:82).

The evolution of this model into its current form is documented in the work of Kabrisky's students. Radoy demonstrated the validity of the Fourier Transform-based process by classifying simple patterns based on a cross-correlation of their low frequency spatial harmonics in the transform domain (Ref 22). Tallman expanded Radoy's work and incorporated an adaptive low pass spatial filter into the model (Ref 27). Gill explored the effect of scale changes on the discrete Fourier Transform. He developed a scaling technique based on the second moment of area, and extended Tallman's work to include pattern scaling (Ref 8). Ginsburg, in a search for psychological correlates, applied the algorithm to a wide range of input patterns. He concluded that the Gestalt principles of proximity, similarity, and closure, as well as many geometric optical illusions, can be explained in terms of the spatial frequency processing mechanisms incorporated within the model (Ref 9).

Figure 3 presents a simplified machine implementation of the Kabrisky model of the human visual system. The

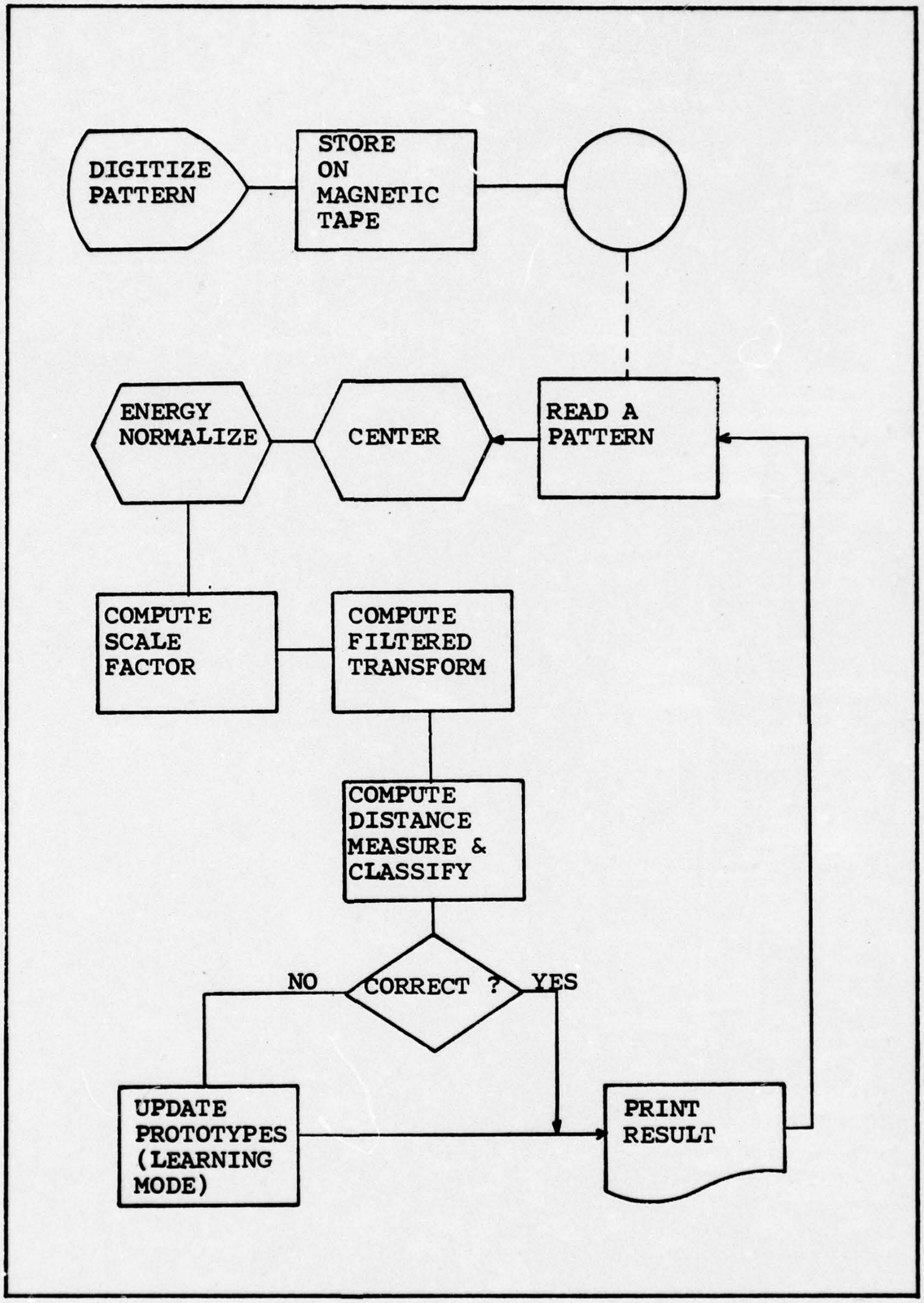

Fig. 3. Computer Implementation of the Kabrisky Model

essential features of this process have been incorporated into a general purpose pattern recognition system. This system serves as a vehicle for research into a wide variety of pattern recognition problems.

Images are digitized by a flying spot scanner and stored on magnetic tape. The digitized images are read from magnetic tape and undergo a preprocessing phase which consists of centering, energy normalization and scaling. Digitization, centering, and energy normalization are equivalent to processes carried out within the eye. The scaling process will be considered in detail later. The transform of the processed pattern is then computed utilizing a two-dimensional fast Fourier Transform algorithm. The transform is spatially filtered and DC normalized. This processed transform can be thought of as a point in N-dimensional space, where N is equal to the number of coefficients in the filtered transform. The classification process requires computation of the Euclidean distance between the point representing the input pattern and those representing stored prototypes. The input pattern is classified on the basis of the closest prototype. The system incorporates an initial learning phase during which the location of the prototypes and the appropriate boundaries between them are established.

Of special interest is the scaling portion of the algorithm. It is, of course, necessary only if pattern

sizes are allowed to change. The Gill algorithm requires an a-priori knowledge of the average size of the patterns being processed. The scale factor is computed by comparing the size measure of the current pattern with this average. The scale factor is subsequently incorporated into the transform computation. This technique has the advantage that it works. However, in terms of modeling the human system, it lacks appeal because of the requirement for an a-priori knowledge of average pattern size prior to the classification process.

The Discrete Fourier Transform

The discrete Fourier Transform is representative of a variety of transformation processes that could conceivably be carried out within the visual cortex. The essential characteristic of the transform-based system is that spatial frequency becomes the media by which the message of the pattern is conveyed. The spatial frequency components are extracted and ordered by the transform. This formatted representation then serves as input to further stages of the recognition process.

The input to the human visual system can be considered as a two-dimensional pattern of finite extent. This pattern is sampled at discrete points. The values of the pattern intensity at each sampled point are weighted and summed to form the discrete Fourier Transform. Although the discrete Fourier Transform is logically derived from the

continuous case, and, indeed, the two functions share similar properties, the discrete Fourier Transform need not be considered to be an approximation of the integral transform.

Definition. Consider an arbitrary discrete spatial distribution a_{mn} for m=1...M and n=1...N. It is convenient to specify M and N to be odd integers. The two-dimensional finite, discrete Fourier Transform may then be defined by the following transform pair:

$$A_{pq}=\sum_{m=1}^{M}\ \sum_{n=1}^{N} a_{mn}e^{-j2\pi(\frac{mp}{M}+\frac{nq}{N})} \qquad (1)$$

$$a_{mn}=\frac{1}{MN}\sum_{p=-P}^{P}\ \sum_{q=-Q}^{Q} A_{pq}e^{j2\pi(\frac{mp}{M}+\frac{nq}{N})} \qquad (2)$$

where 2P+1=M, 2Q+1=N; and $\frac{2\pi nq}{N}$ and $\frac{2\pi mp}{M}$ represent spatial frequencies (cycles per unit length) in the transform plane.

The Scaling Property. The properties of the discrete Fourier Transform have been described in terms of their relevance to the pattern recognition process by Tallman and Gill in references cited earlier. In particular, Gill addressed himself to the scaling property. He showed that for the continuous case, the transform of a pattern uniformally contracted by a factor k is equal to the uniformally expanded transform of the original pattern multiplied by a factor of $1/k^2$. Likewise, the transform

of an expanded pattern is proportional to the contracted transform of the original pattern. This effect is shown for the one-dimensional case in Fig. 4. In one dimension the scaling factor becomes $1/k$ instead of $1/k^2$.

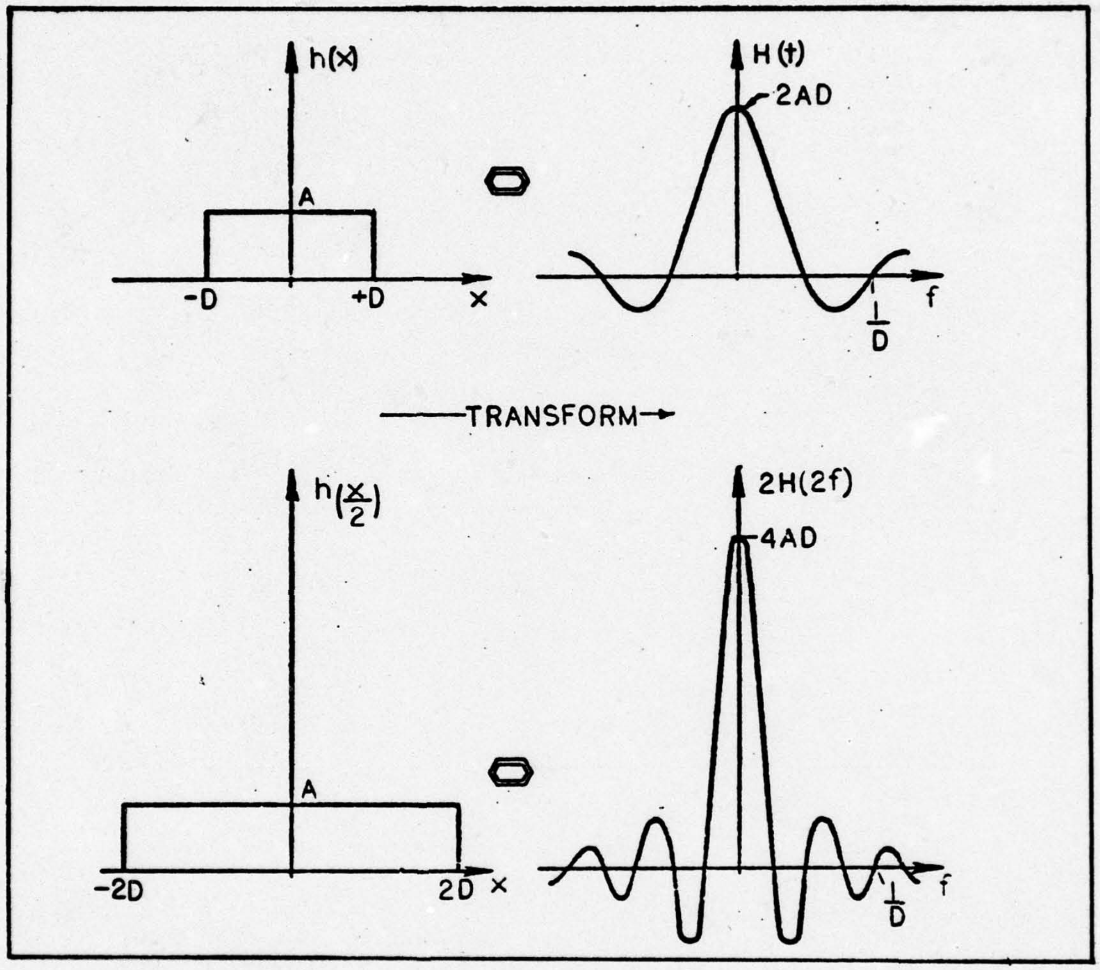

Fig. 4. The Effect of Scale Change

A complication is introduced in applying the continuous results to the discrete case. It arises from the fact that although size transformation is a linear process in

the transform domain, the discretization of a pattern may produce nonuniform effects which change the form of the sampled versions of the scaled pattern. This effect is illustrated in Fig. 5. It is evident that distortion introduced by the sampling process increases as the granularity of the sampling grid relative to the pattern dimensions becomes larger. An additional complication is introduced when the scaling process reduces a pattern to a point where the Niquist sampling criteria is no longer satisfied. The effect will be apparent in the loss of high frequency components in the scaled pattern.

Neither of these effects are relevant to the study of the normal operation of the human pattern recognition process. Patterns small enough to interact with the retinal mosaic in this manner are not a part of normal human visual activity. Furthermore, the resolving power of the optics imposes a lower limit of approximately 11 microns on the resolved image of a point source. The average diameter of cones in the fovea is 1.5 microns (Ref 12:612). Thus many receptors contribute to the discretization of even the smallest resolvable image, and any reasonable pattern will be sampled by such a large number of receptors that the grid will be very fine relative to the dimensions of the pattern. This is the siutation in modern shadow-mask television receivers. The television image is made up of discrete points, however,

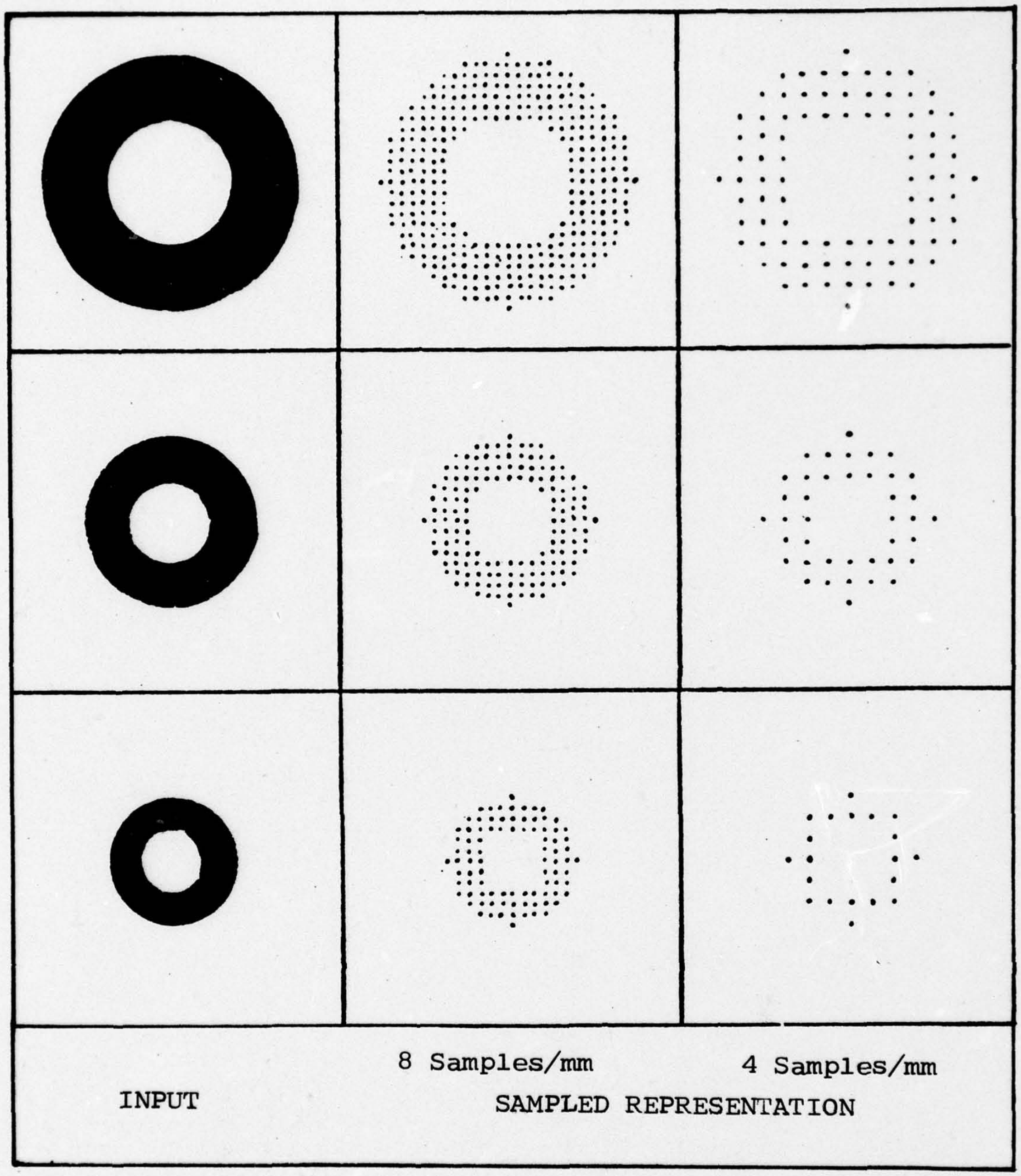

Fig. 5. Distortion Produced by Sampling Granularity

the grid is fine enough that there is no apparent distortion introduced when images are rotated, expanded, or contracted.

Assuming that the sampling interval is small enough so that distortions due to granularity can be neglected, then the results for the continuous case can be applied to the discrete transform. Reiterating the scaling property: expansion (contraction) results in a transform which is proportional to the contracted (expanded) transform of the original pattern. In terms of the transform-based pattern recognition system, the scaling property implies that similar patterns differing only in size will not produce matching transforms. Therefore, the filtered-transform technique alone cannot account for scale invarient pattern recognition. Additional mechanisms must be incorporated into the model if this feature is to be realized. A variety of techniques will solve the problem of scale change. However, if the Kabrisky model is to remain a valid representation of the human visual processes, inferences derived from the incorporation of any specific mechanism must agree with human psychological data. This restriction serves to severely limit the field of possible techniques.

Size Invariant Processing

Whenever a stimulus evokes a single response, it is being "recognized". The problem for theory to explain is

why stimuli differing significantly in size are equivalent in producing the same reaction. This appears to be the case, at least for over-learned symbol sets such as alphabetic characters. The approach taken in the remainder of this section will be to suggest two mechanisms that could account for size invariant processing. Each process will be examined from the viewpoint of its compatability with the filtered-transform model as well as in terms of the possibility that it is utilized within the human visual system. Inferences regarding measurable effects which may be used to differentiate between the two hypothetical implementations will be drawn.

The first mechanism is referred to as multiple prototype storage. This hypothesis assumes that no size scaling or normalization is carried out within the visual system. Therefore, transforms of patterns differing in size will not match. That is, coefficients corresponding to the same spatial frequencies will not be equal. The clustering of points in transform space will no longer be a measure of similarity between patterns because it will have been corrupted by the size variation. This implementation, therefore, requires the storage of multiple prototypes based on the size variation between patterns in the same class. This scheme may initially seem impractical because it appears to require an inordinate number of prototypes corresponding to each class.

However, this is not the case. The filtered transform mechanism itself allows for an approximately fifteen percent size differential between the input pattern and the corresponding stored prototype before it fails (Ref 8:77); this effect is caused by the blurring introduced by the low pass filter which reduces the requirements on precise form congruence between the prototype and the input pattern. It implies that a new prototype would be required only for size changes exceeding thirty percent. Assuming normal images range between visual angles of 1 arc minute, imposed by the limit of visual acuity, and 2 degrees, imposed by the size of the fovea; then only seven prototypes would be required to store the total range of any pattern.

The multiple-prototype hypothesis is supported by the manner in which the human visual system appears to account for the effect of image rotation. Humans do not naturally recognize objects in unfamiliar orientations. Faces of even very familiar people are difficult to recognize when inverted. It is difficult to read inverted text; however, this skill can be learned. On the other hand, a wide variety of objects that people are used to viewing in different orientations, such as tools and eating utensils, are instantly recognized regardless of orientation. It is reasonable to assume that prototypes representing these rotated forms have been stored.

What sort of evidence would support the hypothesis

that the human visual system stores prototypes based on size? The recognition system should fail for patterns that do not correspond to a stored prototype. The failure of the primary system can be expected to result in a "fall back" to a secondary system involving a detailed analysis of the structural components or other features of the pattern. This secondary processing can reasonably be expected to require additional time. A properly designed experiment measuring classification reaction time should disclose this effect.

The second possible mechanism will be referred to as adaptive scaling. The adaptive scaling hypothesis requires that at some point within the visual process, a size normalization is carried out. The scaling may be applied prior to the transform or may be incorporated into the transform calculation itself as suggested by Gill. The final result is that each class is represented by one prototype rather than by multiple prototypes based on size. The impact of this procedure is to reduce storage requirements at the expense of increased computational complexity. The central problem with this approach is to determine the inputs which control the scaling process. Are these inputs some direct measure of size, such as the second moment of area, or some less obvious factor such as apparent distance? This problem will be explored in detail later. A second problem that is relevant to

confront at this point is this: Does scaling precede recognition or follow it? If the filtered transform model is valid, scaling must precede recognition. Since recognition is equivalent to measuring the correlation between transforms, any distortion introduced by size variation must be corrected prior to the recognition process. The determination of the appropriate scale factors cannot be dependent on the recognition of the object, since this would introduce a circular dependency into the recognition algorithm. A state in which inappropriate scaling locked out recognition would then be possible.

Assuming the adaptive scaling hypothesis is valid, what can be inferred about the operation of the human visual system? The scaling process is equivalent to a compression in the degree of size variation. Thus, with the scaling mechanism in operation, it is reasonable to expect that perceived size change will not vary as greatly as the actual size change of the retinal image. Secondly, if the control inputs to the scaling mechanism can be artificially manipulated, then a corresponding distortion of perceived size should result. And finally, if the normalization mechanism takes a finite time, and operates serially within the visual process, then the time required to scale an input pattern should be measurable.

The evidence supporting these two dissimilar hypotheses will be discussed in Chapter IV.

III. Visual Perception Analysis Facility

Introduction

A general purpose, computer-based, visual perception analysis facility (VPAF) was developed as part of this research project. VPAF was designed to replace the commonly used, and less flexible, multi-channel tachistiscope. In addition to display generation, the system was used for recording and analyzing experimental data. The paragraphs that follow briefly outline the major features of VPAF. Detailed hardware schematics are contained in Appendix A. The VPAF software is described in Appendix B.

System Requirements

The requirements for VPAF were derived from the need for a general purpose flexible display generation and presentation system. The system requirements describe these general requirements in more specific terms.

1. The system will be capable of generating a series of two-dimensional displays on a television monitor.

1a. The experimenter will be able to predefine each display by input to the system prior to the experimental run.

1b. Stimulus duration may be preset by the experimenter or controlled in real-time, based on the subject's response.

1c. Stimulus rate may be set by the experimenter or controlled in real-time, based on the subject's response.

2. A response unit will be used by the subject to respond to displays.

2a. Four pushbutton switches will be provided on the response unit. The role of the switches in the control of the experiment may be modified by changes to the software.

2b. Subject responses may be used to modify the experimental sequence in real-time.

2c. A system clock will be provided to measure subject response times to an accuracy of ten milliseconds.

3. The experimenter may specify, by modifications to the software, the data to be recorded or output during an experimental run.

3a. Response data may be stored in memory for post-run printout or analysis. Data may be printed out after each response.

3b. Output format may be specified by the experimenter.

3c. Response data will include, but should not be limited to: display identification, response, and response time.

4. The system will include a flexible, multi-level, software structure.

4a. The experimenter will be able to modify system control algorithms in order to implement a variety of experimental paradigms.

4b. A high level language capability will be available to

provide a means of generating data analysis programs.

5. The system will incorporate, to the maximum extent possible, existing equipment and systems.

System Hardware

The major components of the VPAF system are shown in Fig. 6. The system incorporates a Data General NOVA-2 minicomputer. The NOVA-2 is a 16-bit machine with a 12k core memory. The memory is expandable to 32k words. This computer, and the associated peripherals, was available in the AFIT digital logic laboratory. The peripherals included a paper tape reader, paper tape punch, teletype, and line printer. Of these, only the teletype was essential for program development and control of the experiments. However, use of the other peripherals significantly decreased the time required to develop and modify the software.

Display System. The display portion of the system consisted of a 21 inch television monitor. The video signal supplied to this monitor was generated by a South West Technical Products Corporation CT-1024 Terminal System. This terminal stores and displays two individual pages of 16 lines by 32 alphanumeric characters each. Plug-in modules provide for curser control, display blanking, and serial ASCII communication with a computer. The terminal required minor modification to work in the VPAF configuration. The modification is described in Appendix A.

Two types of displays are generated by the CT-1024

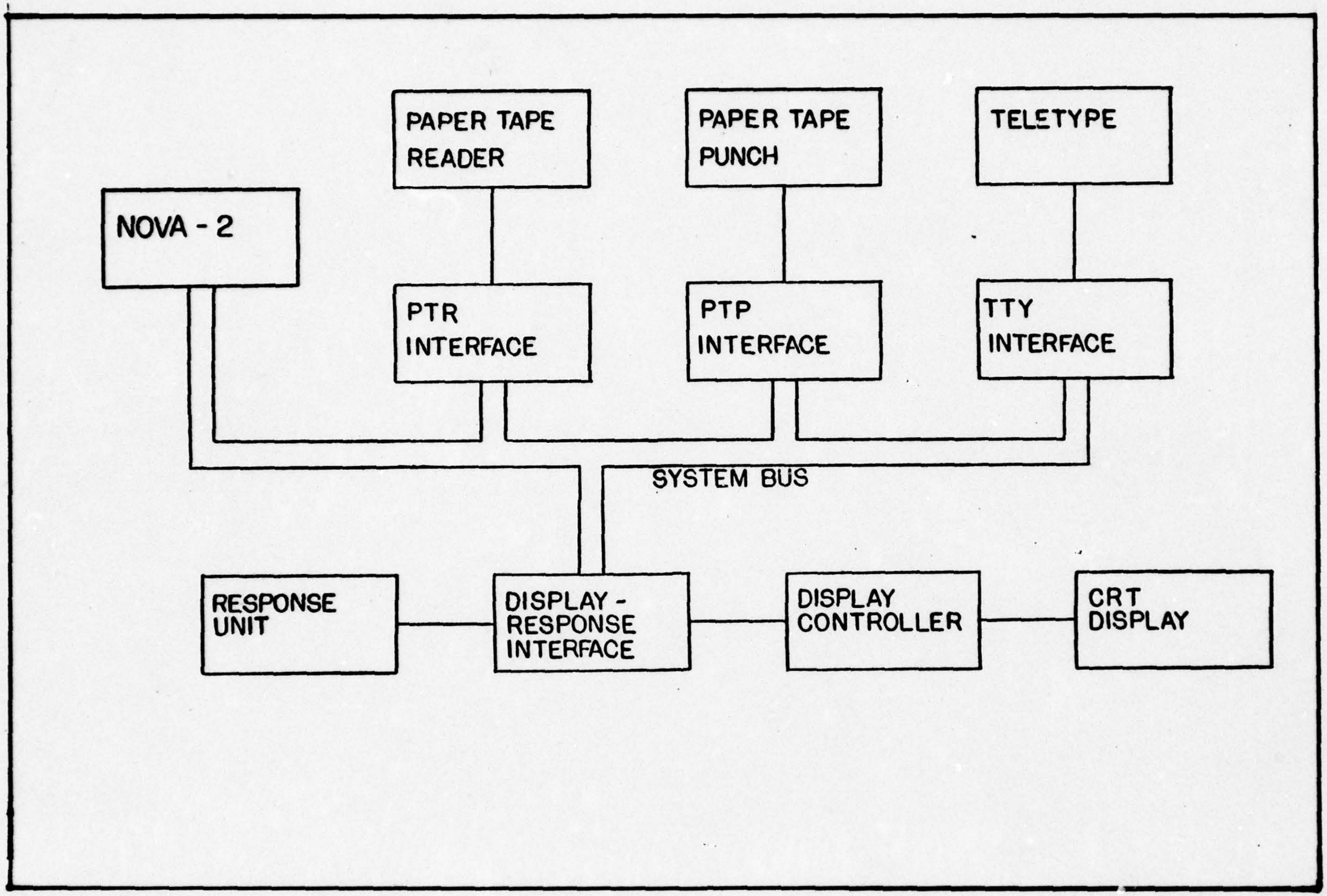

Fig. 6. The Visual Perception Analysis Facility

terminal. In the normal display mode, single alphanumeric characters or lines of text were displayed. In this mode each character consisted of a 5 by 7 dot array. Characters were 1 cm high by 1 cm wide. Vertical separation between lines of text was 5 mm. Horizontal spacing between characters was 2mm. In the second display mode, larger, more complex symbols were built up from groupings of single characters. In this format the entire display was considered to be a 16 by 32 cell array. Symbols were formed by filling appropriate groups of cells. Any of the simple characters could be used as the building blocks of the more complex forms. In practice, the symbol " # " was found to be the most suitable. By slightly defocusing the monitor, this symbol became indistinguishable from a small rectangle.

Response Unit. The VPAF response unit consisted of a metal chasis with two pushbuttons; these were operated by the subject as he viewed the display. Two additional pushbuttons were connected remotely to the response unit to allow the experimenter to control the experiment without interfering with the subject.

Display-Response Interface. The display and response units required special purpose interfaces with the computer. In order to minimize chip count, the common functions required by the two units were merged into one interface. This increased the complexity of the status information

circuits, but avoided duplication of the basic control circuits.

The system utilized status bits to specify which unit required servicing. I/O processing was carried out on an interrupt driven basis. An interrupt was generated when the display controller required a new character or control information, or when a response had been made. These requests generated an interrupt as well as providing a unique device identifier to the processor. The device address was decoded by the software to allow program execution to be transferred to the software modules that controlled the display-response interface. A further, more detailed, description of the interface is contained in Appendix A.

System Software

The system required a broad spectrum of software to provide for the implementation and modification of experimental procedures as well as the development of data analysis programs. The software available in the VPAF system is shown in Fig. 7.

An advantage of basing VPAF on a mature product line, such as the NOVA system, was that a great deal of manufacturer supplied software was available. Utilizing this software resulted in a significant reduction in development time. The essential tools such as editors and assemblers were immediately available. The full development effort could, therefore, be devoted to the relevant system control software.

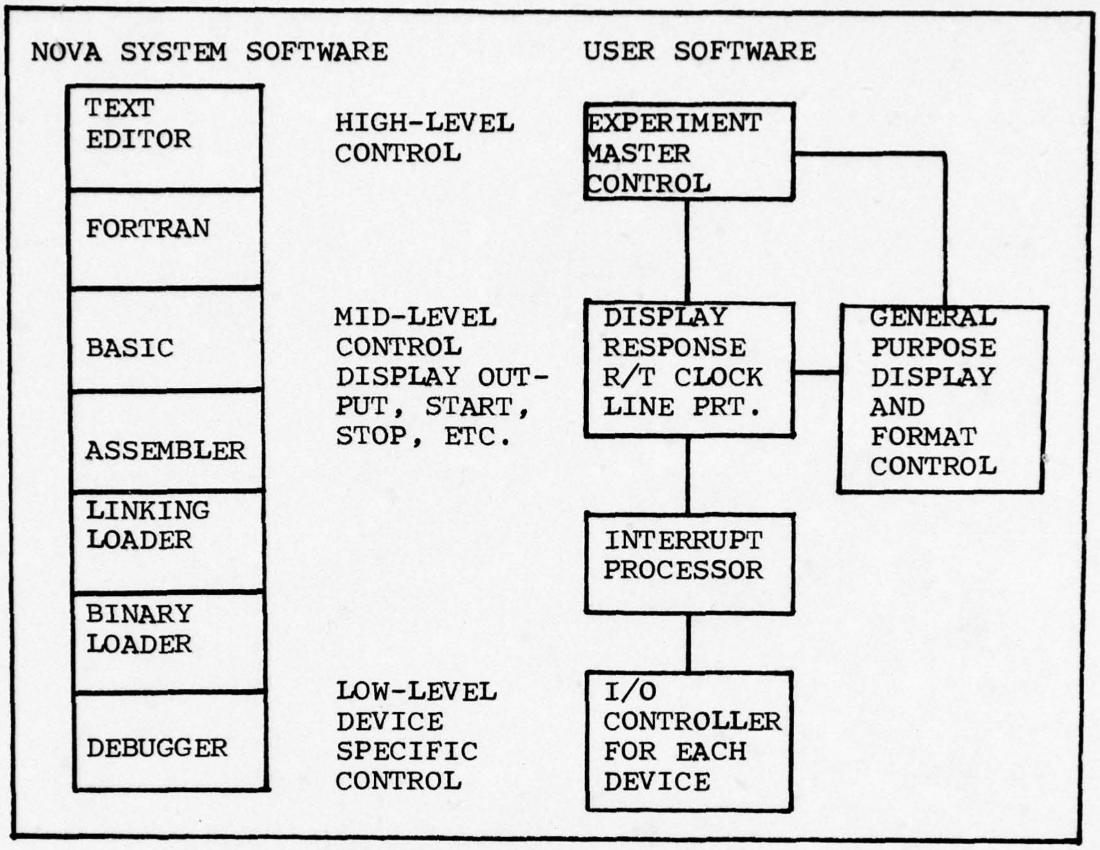

Fig. 7. VPAF Software

VPAF source programs could be written in assembly language, BASIC, or FORTRAN. The text editor was used to create the initial source programs that served as input to the assembler or FORTRAN compiler. A relocatable object tape was produced for each software module. The relocatable modules were loaded and linked by the relocatable loader to produce an executable version of each program. This procedure was used to develop the device control modules, as well as the experiment control software.

A multi-purpose experiment control operating system

resided in core during an experimental run. This program was interrupt driven. It idled until an interrupt from an I/O device was received. It then passed control to the appropriate hardware control modules. The system contained modules that controlled the display system, response unit, teletype, line printer and the real-time clock. Each of these devices could be active during a run.

A set of multi-purpose display modules also resided within the system. These modules allowed the experimenter to define displays and control the display duration and rate.

The experimental sequence was specified by a series of calls from the main program control loop to the display control modules. This method allowed the experiment to be modified by changes to the main program, rather than to the many lower level modules.

Post run data analysis programs were written in FORTRAN, or executed directly from the teletype through use of the BASIC interpreter. The BASIC facility provided the means to quickly develop and run statistical analysis routines. When more powerful data manipulation was required, the programs were written in FORTRAN.

IV. Experimental Support for Adaptive Scaling and Multiple-prototype Storage

Introduction

In this chapter the experimental evidence supporting the adaptive scaling and the multiple-prototype hypotheses will be examined. Although the term adaptive scaling has not previously been applied to describe the phenomena known as size constancy scaling, these terms appear to be equivalent descriptions of the same mechanism. Therefore, the literature concerning size constancy is relevant to this study and the results of several significant experiments in this area will be reviewed. It will be shown that the mechanism that provides for size constancy is equivalent to the adaptive scaling mechanism suggested by inferences drawn from the Kabrisky model. Little work of relevance to the validation of the multiple-prototype hypothesis has been reported in the literature. Therefore, a series of experiments utilizing the VPAF system were devised to test the multiple-prototype hypothesis. These experiments will be described in detail.

Before proceeding further, it is necessary to develop a precise definition of size and size change, since there is a great deal of potential confusion in these terms. The actual physical size of a stimulus object or pattern will be referred to as physical size. An object in the

visual field subtends an angle that is a function of both physical size as well as the distance between the object and the observer. The visual angle is a direct measure of the size of the projected retinal image, which will be referred to as retinal size. It is obvious that retinal size alone is a poor indicator of physical size, since different physical sizes at different distances may result in equal visual angles. Finally, size as indicated by the response of the observer, will be referred to as perceived size. In terms of the human visual process, retinal size is one input to the system. However, retinal size alone is ambiguous; additional inputs are required for an accurate judgement of physical size. These additional inputs will be considered in the following section.

Size change will be defined as a change in the size of the retinal image. This is equivalent to a change in the visual angle subtended by the object of interest. Size changes may be produced by changing the physical size of the stimulus or, alternatively, the distance between the observer and the stimulus may be varied. The visual environment is in constant motion in three dimensions. Visual angles are, therefore, constantly changing, producing continuous changes in the retinal size of each object in the field of view. It is possible that adaptive scaling has evolved as a result of the complex visual environment.

Evidence of adaptive scaling in infants (Ref 3) supports the conclusion that the scaling mechanism may be hardwired into the visual system.

Adaptive Scaling

The perceptual system transforms retinal images in systematic ways according to perceived distance. In general, the transformation tends to maintain perceived size more or less constant in spite of changes in viewing distance. The end result is known as size constancy. Size constancy is a special case of the general concept concept of visual constancy, which applies to brightness, shape, and other perceived characteristics in addition to size.

Size constancy studies have converged upon a single proposition which formalizes the relationship described above; it is aptly called the size distance invariance hypothesis. It states: A retinal image, or visual angle of a given size, determines a unique ratio of perceived size to perceived distance. Another formulation of this proposition is known as Emmert's Law. It states: The apparent size of an object will be proportional to apparent distance when retinal size is constant. Emmert's Law has been employed in the investigation of the perceived size of an afterimage and its relationship to the distance of the surface onto which it is projected.

Support for the invariance hypothesis comes from studies which show that size of an unfamiliar object can

be judged accurately only if cues to the distance of the object are available.

Scaling Experiments. Thouless was the first experimenter to carry out detailed experiments designed to measure the size constancy effect, as well as to determine the relevant scaling control inputs (Ref 29). Thouless measured size constancy by comparing a cardboard shape, of a circular disk for example, placed at a given distance with a series of disks of different sizes at some other distances. By selecting the closer disk appearing the same size as the more distant one, and by establishing by geometry the difference in retinal sizes of the two disks, an estimate of constancy could be obtained. This estimate can be expressed as a ratio, with perfect constancy giving a value of one and no constancy giving a value of zero. Thouless found that the degree of constancy is a function of the amount of information indicating distance and orientation, as well as personality variables and training in perspective drawing.

Holway and Boring carried out a series of experiments in order to further define the factors that control the scaling process (Ref 14). They obtained size matches under four sets of conditions which represented successive elimination of distance cues. Size matches approximated constancy under conditions of binocular viewing and gradually became a simple function of the visual angle alone

as distance cues were eliminated.

Because the limits of visual acuity are eventually exceeded as the distance between the object and the observer is increased, it is reasonable to ask if size constancy breaks down as the object approaches the distance where it ceases to be visible. It is easy to suppose that an object disappears by becoming perceptually smaller; however, experiments described by Gibson show that this is not the case (Ref 7).

Gibson required observers to judge the length of a 71 inch pole at distances of up to 1/2 mile away. At this distance even a man-sized object is difficult to make out. He concluded that under favorable conditions for determining distance, an object can be seen with approximately its true size as long as it can be seen. Its size does not become smaller, but only more indeterminate.

The results of experiments such as these suggest that there is no such thing as an impression of size apart from an impression of distance. An adaptive scaling front end on the visual processor is constantly in operation. Its effectiveness is determined by the availability of the necessary control inputs consisting of distance cues. Fortunately, distance cues are available from a variety of sources, both from within the visual scene and from within the visual system itself. These distance indicating inputs work together to set the appropriate scaling factor.

Distance Cues. The distance cues provided by the visual system itself include convergence information and binocular disparity. As an object is moved closer to the observer, the eyes must rotate inward to track it; as an object is moved further away, the eyes swing outward until the optical axes are essentially parallel. The angle of convergence is, therefore, a direct measure of distance. Convergence is an adequate depth cue to approximately 10 feet. Further than this, the eyes are nearly parallel, and the angle of convergence changes little with further increase in distance.

Binocular disparity is a powerful depth cue. In the experiment cited earlier, Holway and Boring showed that size constancy is reduced considerably when binocular disparity is eliminated. The depth cue arises from the fact that when the eyes are converged on an object, there is not a one-to-one correspondence between the retinal images. Each eye views the object from a different angle. Therefore, corresponding receptors in the two retinas respond to slightly different points on the object. The degree of this disparity is interpreted by the perceptual system as a measure of distance, producing stereo-depth perception. The stereo cue is so powerful that properly oriented two-dimensional pictures providing disparite images to each eye produce a three-dimensional perception. After approximately 50 feet, binocular disparity is reduced

to a level at which it no longer provides a valid depth cue. At greater distances there appears to be a wide range of cues associated with the visual scene itself that set the scaling mechanism.

Linear perspective is the primary depth cue within the visual scene. Parallel lines directed away from the observer converge in the distance. Perspective is such an integral part of the real visual world that pictures drawn without perspective seem distorted and inadequate. Textural gradient, motion parallax, height in the plane, interposition, and familiar size have also been shown to be important depth cues. Each of these features has been successfully used by artists to incorporate a suggestion of depth in pictures. Photographs which are, of course, merely two-dimensional representations of the real world, incorporate these depth cues also.

Gregory has suggested that a wide variety of optical illusions are a result of the inappropriate triggering of the constancy scaling mechanism (Ref 11:93). Gregory's theory of inappropriate constancy scaling is controversial, and may in fact not apply to the wide range of illusions he contends it does. However, it does adequately explain effects obtained when depth cues are incorporated into two-dimensional images. Fig. 8 and Fig. 9 provide examples of the effect of inappropriate scaling. In each scene the depth cues incorporated into the illustration serve

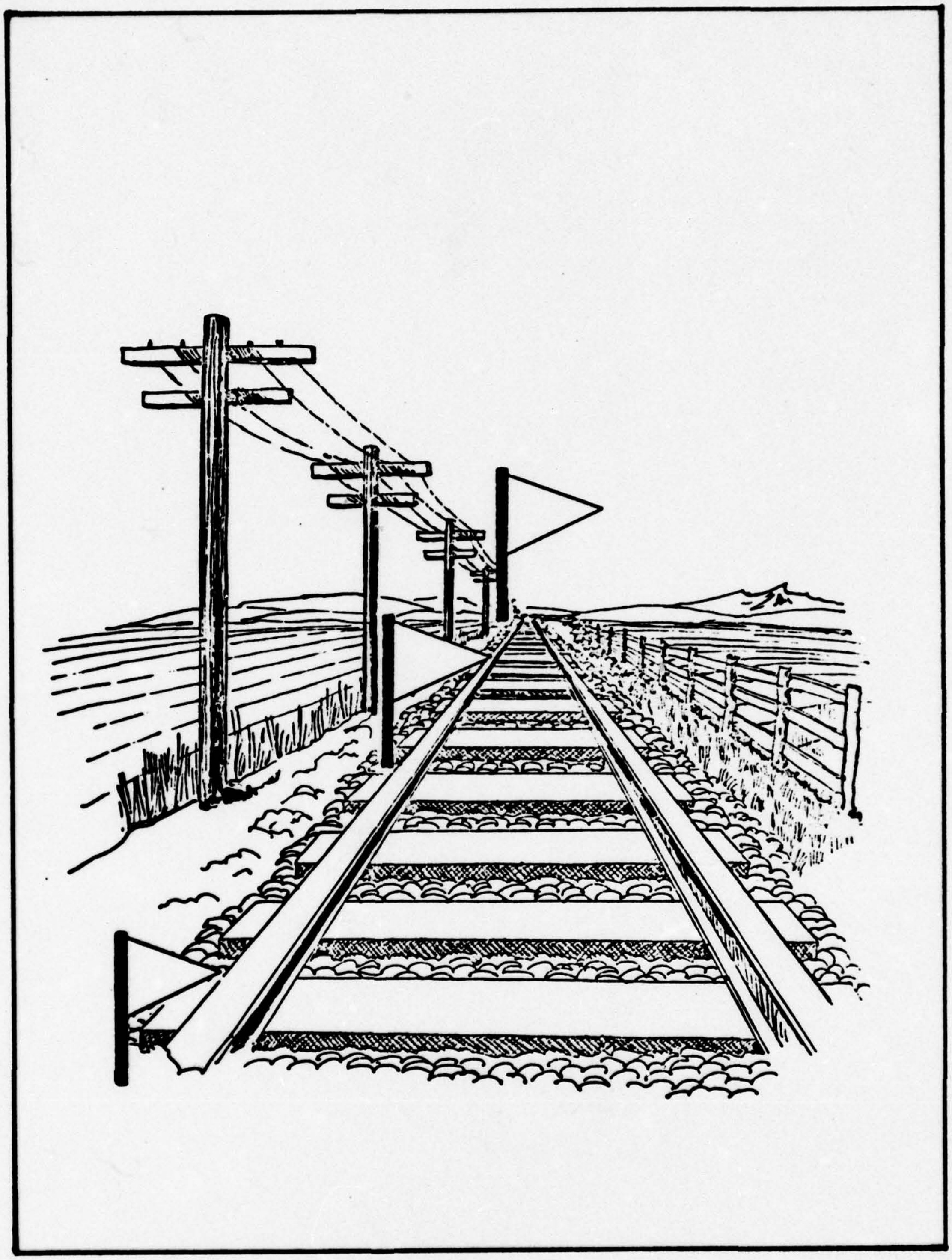

Fig. 8. Size Distortion Resulting From Inappropriate Scaling (From 24:53)

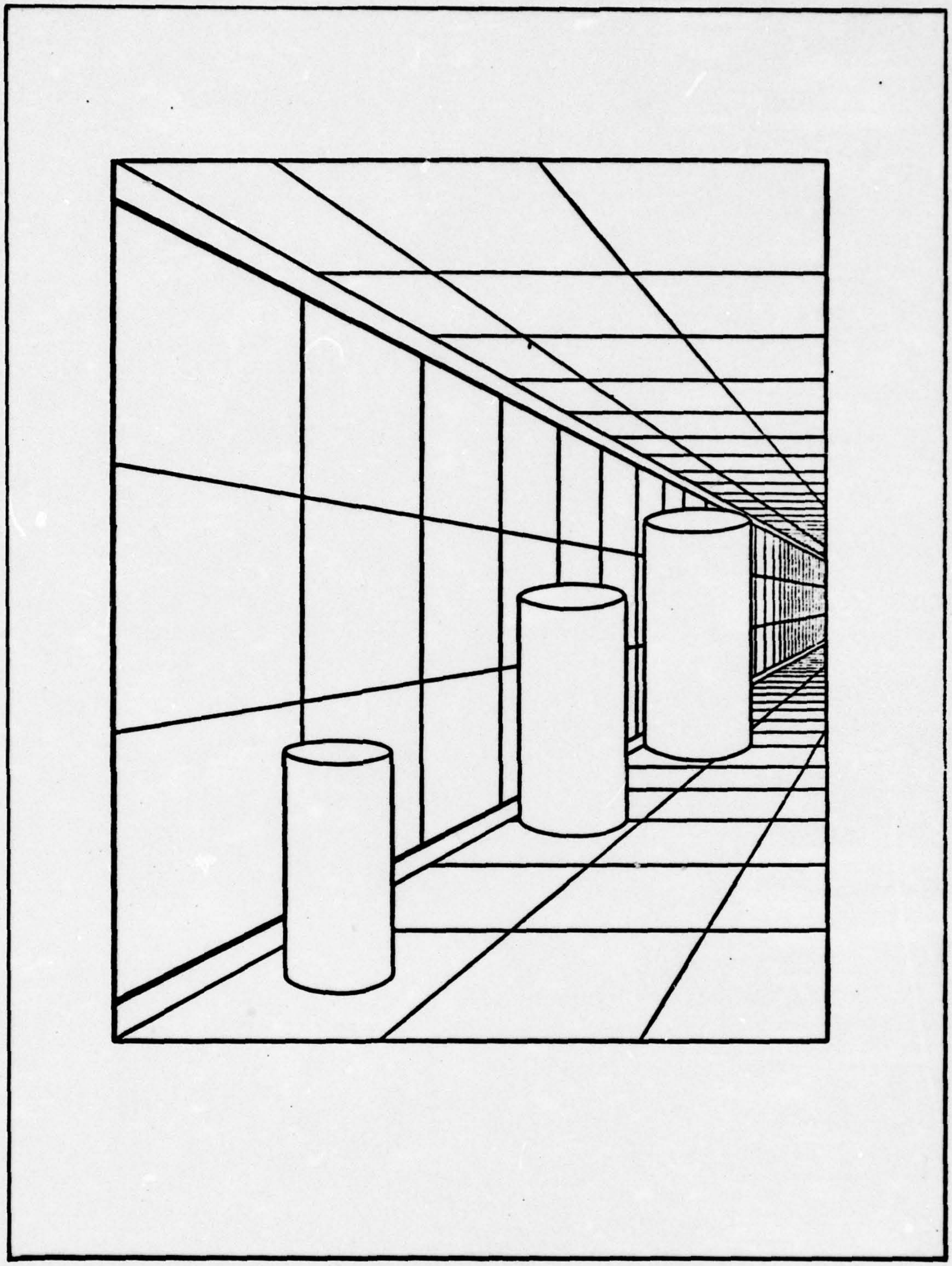

Fig. 9. Linear Perspective and Texture Gradient As Depth Cues

as inputs to the adaptive scaling mechanism which would normally compensate for shrinking size with distance. However, because the illustration is flat, a size distortion is produced by the inappropriate scaling. The more distant objects appear larger than those of equal size in the foreground. If it is the case that depth cues trigger the scaling mechanism, then as depth cues are added to a scene, the distortion due to inappropriate scaling should be intensified. This effect can, in fact, be observed as illustrated in Fig. 10. Each pattern in the series incorporates increasingly stronger depth indicators. The cues result in differential scaling of the symbols incorporated within the figures. The perceptual size difference becomes more apparent as the suggestion of depth is strengthened. Figure 11 is a reversible figure based on the Necker cube. The figure has a distinct shape associated with each stable orientation. When the smaller surface appears closer, the figure is perceived as a truncated pyramid. In the reversed orientation, the figure appears as a rectangular cube. This shape transformation is the result of adaptive scaling. When the smaller surface is perceived as more distant, it is enlarged by the adaptive scaling mechanism until it appears to be the same size as the nearer surface.

It is significant that in all of these illusions, although the observer is aware that he is viewing a

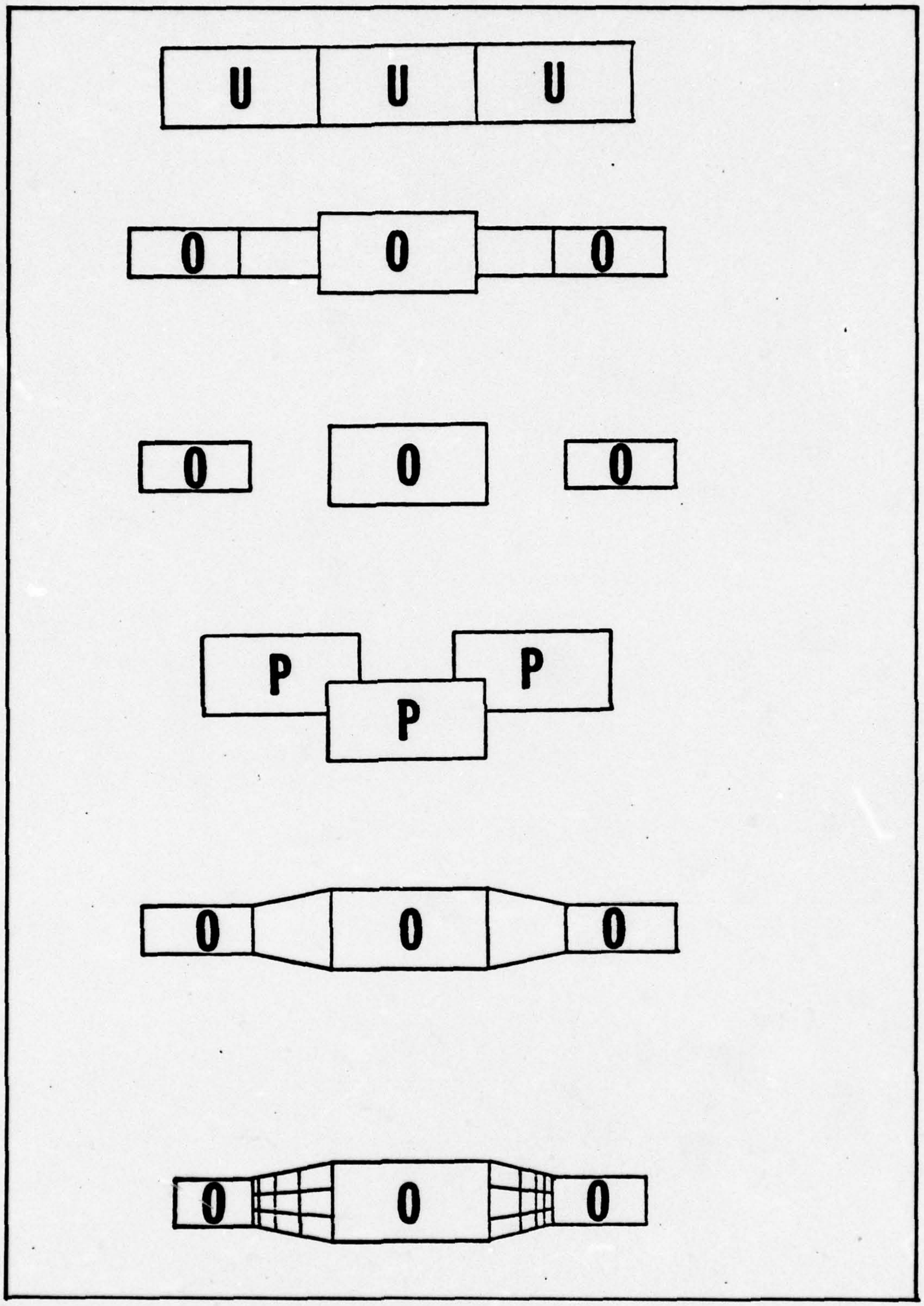

Fig. 10. Additional Depth Cues Affect Scaling

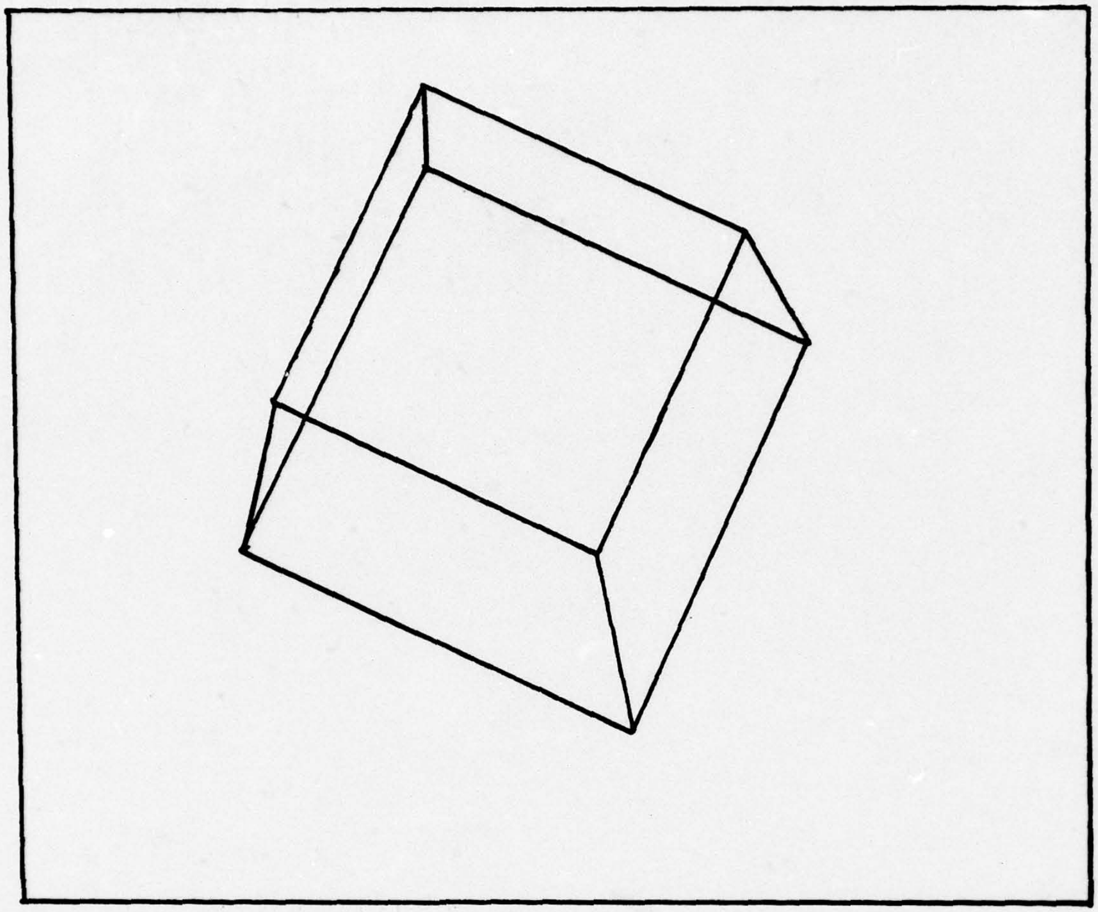

Fig. 11. Necker Figure with Unequal Sides

two-dimensional figure, it is impossible for him to suppress the effect. This provides further evidence that adaptive scaling is a fundamental part of the visual process and might be regarded as a "visual reflex". Humans may learn specific contextual depth cues relative to their environment. Once learned, the cues serve to control the scaling mechanism which cannot be completely suppressed intellectually even though the cues appear in an inappropriate context.

Summary of the Scaling Hypothesis. The evidence obtained from studies of size constancy points to the existence of adaptive scaling within the visual system. Scaling is controlled by distance information. Setting the scale based on distance cues is a process which is independent of the recognition process. This is one of the primary requirements of the adaptive scaling hypothesis. In practical terms, it implies that when reading these words, the scaling factor is cued by the distance from the page to the eye, not by the system recognizing the symbols as printed text and then setting the scale to make them an appropriate size.

The success of the adaptive scaling hypothesis in accounting for size changes due to distance change does not rule out the need for other mechanisms. Scaling does not provide an explanation of how stimulus objects of unequal size at equal distance from the observer are correctly classified. As stated previously, the filtered transform technique breaks down when the size of the input pattern varies by more than 15 percent from the size associated with the stored prototype representing the class. Therefore, the filtered transform model alone cannot account for recognition under these circumstances either. The multiple-prototype hypothesis suggests a mechanism to fill this gap in the model. For this reason, a series of experiments was devised to test the validity

of this hypothesis.

Tests of the Multiple-prototype Hypothesis

The first question to be answered was whether or not the human pattern recognition system does, in fact, fail in the situation described previously - change in the size of the retinal image without a corresponding change in distance. There are theories of pattern recognition, such as those based on geometric feature analysis that imply that the size of the pattern (within the limits of foveal regard) should have no effect on the recognition process. Therefore, the first experiment was designed to detect if what will be referred to as uncompensated size change causes a "glitch" in the pattern recognition process.

Experiment One. The multiple-prototype hypothesis suggests that learning to recognize a pattern at a specific size will not provide for equivalent recognition of the pattern at all other sizes. This prediction served as the basis for the paradigm. The subjects were required to learn to recognize the members of a set of abstract symbols. Each subject was then tested on a set consisting of scaled versions of the symbols in the original set. The test set consisted of symbols enlarged by 100%, reduced by 100%, and equal in size to the original symbols. Reaction times required for recognition were measured for learned (old) sizes and unlearned (new) sizes.

The task utilized for this experiment was an adaptation of what Sternberg called the character-classification task (Ref 26). This task is a generalization of the simple recognition task. Symbols are presented sequentially as test stimuli. The subject is required to make a positive response if the stimulus is a member of a small "target" set called the positive set, and a negative response otherwise. The subject is instructed to respond as rapidly as possible while maintaining a low error rate. Sternberg used this technique to study high speed scanning in the human memory (Ref 25). His results showed that the classification response time is directly proportional to the number of elements in the positive set, increasing approximately 35 milliseconds for each additional element.

In this experiment, the VPAF system was used to generate two sets of abstract symbols. Each symbol in the second set was formed by scaling a symbol in the first set by a factor of two. The smaller symbols were 60 mm by 48 mm; the larger were 120 mm by 96 mm. Each symbol was centered in a 25 cm by 40 cm rectangular window positioned 2 meters from the test subject. The two sets of abstract symbols utilized in the experiment are illustrated in Fig. 12.

The test subject held a push-button in each hand. Each subject was instructed to respond to the positive set by depressing the right-hand switch and indicate a negative response with the left hand push-button. The

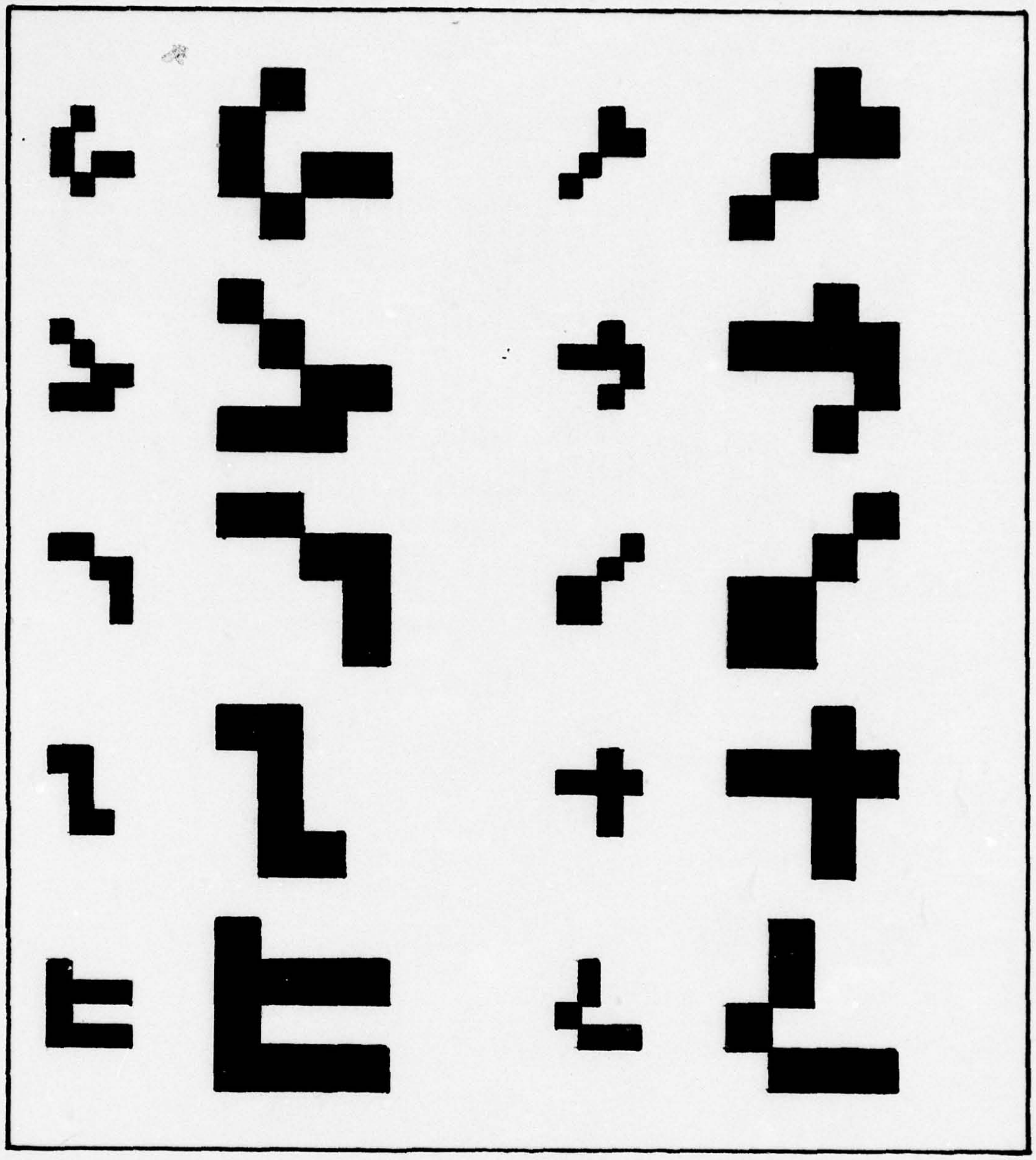

Fig. 12. Experiment One Symbol Set

subject was instructed to ignore size when making the classification. Three symbols were preprogrammed as the target

set. There were, therefore, six correct positive responses since each of the symbols appeared in two sizes.

Each stimulus was displayed for 100 milliseconds. Response time was measured from the time the display was enabled until either push-button was depressed. The first switch closure locked out further responses so that the first response was always recorded. The clock within VPAF was utilized to measure response time to the nearest 10 milliseconds. Between each display response sequence, the response time and other relevant information were printed by the VPAF line printer. The printing operation required 4 seconds, at which time a new display was initiated.

During both the practice and test phases of each experimental run, the symbols were randomized with respect to size and target set membership. Five large and five small symbols were displayed during the practice mode. No practice symbol was displayed at both sizes. The subject was able to become familiar with the equipment and experimental procedure including the size variation aspect while learning to recognize only one size of each symbol. Ten unlearned symbol-size combinations were available for use during the test phase.

During the practice period, the feedback message "CORRECT" or "WRONG" was displayed after each response. The subject was required to determine the members of the

target set by trial and error. After 20 consecutive correct responses, the test phase was initiated. In the test mode, no feedback messages were displayed. In several pilot studies conducted to refine the paradigm, it became evident that subjects tended to over-react to mistakes. Their frustration with an error adversely affected their performance on the following stimuli. This effect was reduced by deleting the feedback during the test. The test sequence consisted of a series of 20 stimuli. Each symbol was displayed at two sizes. Ten of the sizes were viewed for the first time; ten had been learned during the practice period.

Five male graduate students with normal or corrected vision participated in the experiment. Mean reaction times for each subject were calculated for large and small symbols within the learned and unlearned groups. Error rates ranged from 0 to 15 percent for the learned sizes, and 5 to 30 percent for the unlearned sizes. The incorrect responses were discarded prior to calculating the reaction time means. The average reaction times across subjects and sizes for the learned and unlearned groups is shown in Fig. 13. The learning effect was significant ($F(1/4)=9.54$, $p<.05$). The effect of size was not significant ($F(1/4)=2.28$, $p=.21$). The means for large and small symbols within the two groups were averaged to arrive at the mean reaction time for learned versus unlearned size symbols. The unlearned combinations

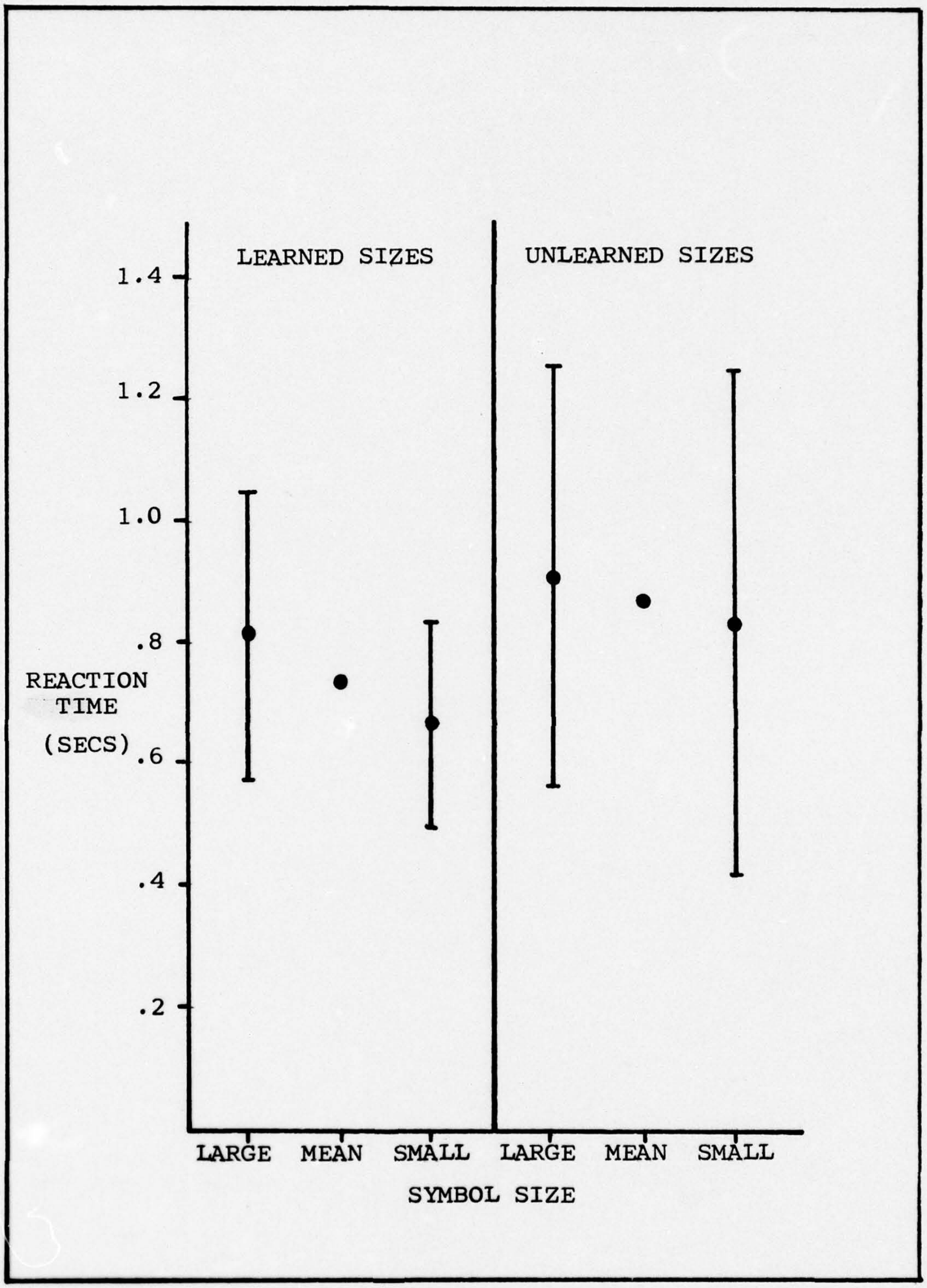

Fig. 13. Reaction Time For Abstract Symbols

required on the average 17 percent, or 128 milliseconds, longer to classify than the learned combinations.

The increased reaction time and the higher error rate for unlearned sizes are results predicted by the multiple-prototype hypothesis. The theory explains the results in terms of the unavailability of stored prototypes for the unlearned sizes. The lack of appropriately sized learned prototypes causes a failure of the primary system and a fall-back to a less efficient secondary process.

The results fail to support the idea that additional scaling or normalization is carried out within this stage of the pattern recognition process. Normalization would result in an equivalent size for all stimulus representations. The recognition process would then function independently of size change, and the system would not have failed on the unfamiliar sizes.

Experiment Two. A second experiment utilized alphabetic characters rather than abstract symbols. Except for this modification, it was similar in all respects to the previous experiment. A second group of five adult males with normal or corrected vision served as subjects.

An assumption underlying the second paradigm was that because alphabetic characters are an overlearned symbol set, they should have a well developed set of prototypes based on multiple sizes. This implies that the practice effect (learned versus unlearned sizes) for

alphabetic symbols will be negligible, or alternatively, the difference in categorization response time for learned versus unlearned sizes should be less for alphabetic characters than it is for abstract characters. This hypothesis was tested by an analysis of the combined results of the first and second experiments.

The results for experiment two were as follows: the mean reaction time for learned sizes was 574 milliseconds; the mean reaction time for unlearned (i.e., unpracticed in the experimental sequence) sizes was 558 milliseconds. This difference of 16 milliseconds, amounting to only 3 percent of the mean reaction time, implies that within the limits of measurement provided by this experiment, there was no significant effect due to practice. This is contrasted with a difference of 128 milliseconds or 17 percent for abstract symbols.

To further explore the significance of the difference in reaction times for learned and unlearned sizes of both alphabetic and abstract symbols, the mean difference in reaction times for each subject in the two experiments was calculated. For the abstract symbols the learned sizes were, on the average, classified 128 milliseconds faster. For the alphabetic characters, the average difference was 15 milliseconds in favor of the unlearned set. This difference is significant at the $p<.05$ level. It implies that categorization of over-learned symbols is

less dependent on practice in a specific experimental arrangement than is categorization of abstract symbols. This is the result predicted by the multiple-prototype hypothesis. Therefore, experiment two, while not proving the existence of multiple prototypes, certainly provides further support for the multiple-prototype hypothesis.

Experiment Three. An experiment was devised to investigate the timing relationships underlying the recognition subprocesses in order to determine if these processes were affected by an uncompensated size change. The first two experiments were primarily concerned with an examination of the relationship between size-familiarity and reaction time. The third paradigm was an attempt to eliminate the familiarity factor in order to more clearly illuminate the effect of scale change.

Sternberg has suggested that the classification task is a two stage process (Ref 26:189). A preprocessor operates on the stimulus to produce an abstracted representation of the symbol. The classifier then makes use of this representation to carry out an exhaustive comparison with the target set. This model is similar to the Kabrisky model where computation of the filtered transform is equivalent to preprocessing, and the comparison task is equivalent to matching the transform to stored prototypes.

Because there is a linear relationship between classification reaction time and the number of elements

in the target set, a regression line through reaction time versus target set size data will indicate the time required to process each additional target set element. The regression line may be extended to intercept the y (reaction time) axis. The point of intercept represents the processing time for those operations common to the classification task, regardless of the size of the target set. This value represents the neuro-mechanical reaction time of the subject as well as any visual system preprocessing that operates sequentially with the neuromechanical processes. The visual processing time will be masked to the extent that it is carried out in parallel with other operations.

The technique described above was utilized to investigate the effect of size change. The data was analyzed in terms of whether or not a symbol was preceded by a symbol of the same size. Reaction times were grouped into two classes according to their association with change or no change. The results, which will be presented after a more detailed description of the paradigm, showed a significant difference between the two groups of responses.

The experiment consisted of four classification task sequences, each using a different target set. The sets contained one, two, three, or four elements. Each target set symbol was used only once within the sequence of four experimental runs in order to eliminate confusion between

target sets. The four target groups were composed of the symbols: G; A, X; N, C, S, and T, J, D, L. The set of symbols which constituted the negative set for each sequence were: B, E, F, H, I, K, M, O, P, Q, U, W, Y and Z. A ratio of 1:3 for targets to non-targets was used for each of the experimental runs.

Each character was defined within a 5 by 7 array; the array for the large symbols was twice the size of the array for the small characters. The characters were formed on the display screen by " # " symbols placed at appropriate points in the arrays. From a distance of 4 meters, the symbols appeared to be constructed from small rectangles. The technique used for forming the symbols is illustrated in Fig. 14.

```
##########
##########
##
##                    #####
##                    #
##                    #
##########            #####
##########                #
        ##                #
        ##            #####
        ##
        ##
##########
##########
```

Fig. 14. Alphabetic Character Display Technique

At the 4 meter distance used in this experiment, the solid angle subtended by the large characters was $2^{o}4'$ by $2^{o}9'$. Each symbol was displayed for 100 milliseconds. A 4 second delay separated each display from the preceeding response.

Sixteen adult male subjects with normal or corrected vision participated in the experiment. Each subject was tested utilizing each of the four target groups in a counter-balanced design in order to reduce effects generated by the repetitiveness of the experimental runs. As in the two previous experiments, each experimental run consisted of a practice and a test sequence. Feedback messages were provided during the practice session. Both sizes of each symbol were displayed during the learning period in order to minimize any hypothetical effects due to inadequate development of a prototype set.

The mean error rate for all subjects was 2 percent. There was no significant variation in the error rate across target sets. The incorrect responses were discarded prior to computing the means for each subject. Means for each subject were computed for ten samples representing the following size change events: Large to small, Large to large, Small to large, and Small to small. A total of 16 mean reaction times representing 160 samples across target size and scale change were,

therefore, available for each test subject. A three way analysis of variance was carried out on the data for the 16 subjects. The following factors proved to be significant: Number of elements in the target set ($F(3/45)=33.5$, $p<.01$), Change versus no change ($F(1/15)=10.9$, $p<.01$), and size of the stimulus ($F(1/15)=6.8$, $p<.02$). The data for the 16 subjects is presented in Table I.

TABLE I

Mean Reaction Time as a Function of Elements in the Target Set and Size Change

	Elements in Target Set			
Description of Change	1	2	3	4
Small to large	390	431	460	467
SD	33	50	48	52
Large to small	405	438	463	472
SD	39	47	55	52
Large to large	397	417	457	438
SD	35	38	58	61
Small to small	388	414	438	490
SD	29	46	46	67

The effect of size change is shown graphically in Fig. 16. Each data point represents a mean across subjects and sizes. Regression lines have been calculated for each

set of data and are shown on the graph. The correlation between no-change reaction time and target set size was .99; the correlation for the change data was .96.

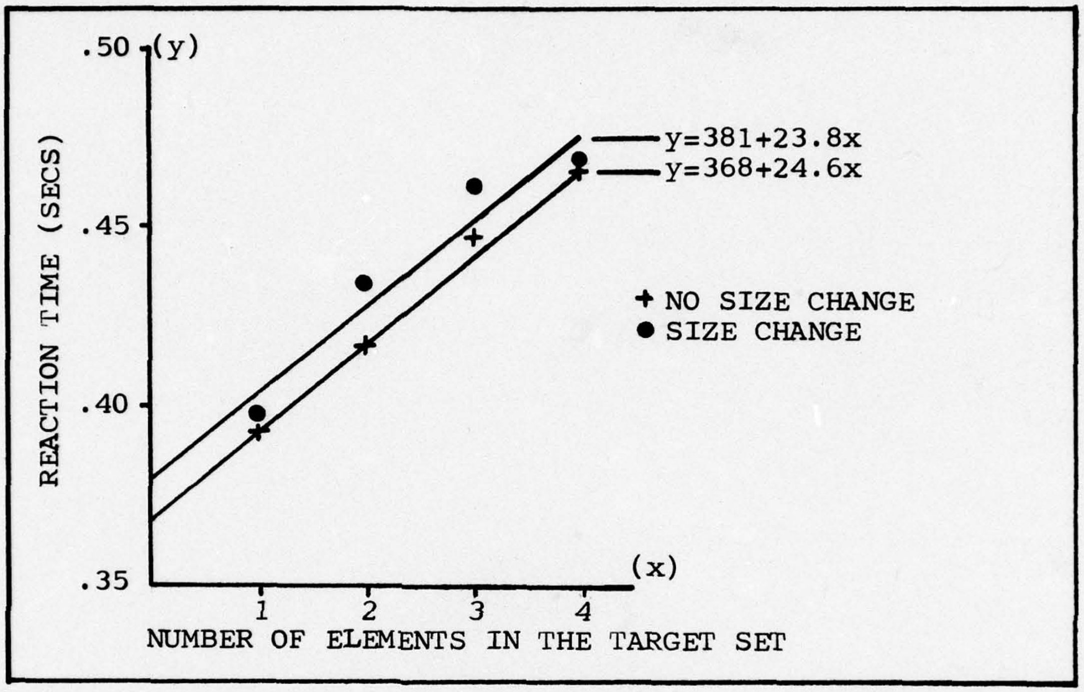

Fig. 15. Reaction Time For Alphabetic Characters

The relationship between these two regression lines provides an insight into the operations carried out within this level of the visual process. Most significantly, the results show that the comparison process is carried out in a manner that is essentially independent of the size change, as noted by the similarity in the slope of the two regression lines. This effect may be explained by assuming that the preprocessor has corrected for the size change in forming the abstracted representation of

the stimulus. Or, alternatively, the comparison process may, in fact, be insensitive to size, as would be the case if multiple prototypes were utilized.

The difference in the y-intercept of 13 milliseconds implies that additional preprocessing is required as the result of the scale change. Although this experiment did not provide enough evidence to adequately determine the nature of this additional processing involved in the classification task, it did serve to isolate the effect of size change to the preprocessing phase rather than the comparison phase. This is an important result in terms of its impact on the Kabrisky model. It must be reconciled with the other experimental results reported in this chapter in order to derive a revised model that adequately accounts for the effects of scale change.

V. Conclusions and Recommendations

Analysis of Experimental Evidence

The size-constancy experiments described in the previous chapter form only a small portion of the total effort devoted to the study of this phenomenon. The effort has provided a broad base of support for the concept of scaling within the human visual system. Distance cues serve as the control mechanism for the scaling process. The cues range from built-in mechanisms, such as convergence, to those relying on higher level analysis of the visual field.

Scaling based on distance, referred to as adaptive scaling in this paper, appears to be an intimate part of the visual process. It is difficult to ignore, or turn off, the scaling mechanism, even in situations when inappropriate scaling produces perceptual distortions, as in the case of the optical illusions described in the text.

Adaptive scaling serves to compress the total range of size differences resulting from the constantly varying distance relationships in the visual environment. It does not, however, apply to scale changes where the difference in retinal size is produced without a corresponding change in distance. Additional mechanisms are required

to account for accurate recognition in the presence of these uncompensated size changes.

Two additional theories were suggested to account for recognition capabilities that apparently operate independently of uncompensated size change. The multiple-prototype theory implies that prototypes based on size are stored as required to represent the range of stimuli sizes. The scaling hypothesis requires that additional scaling be carried out in order to normalize all abstracted representations to a common format prior to the recognition process.

Experiment One provided results that allowed a differentation to be made between the validity of the two theories. The results clearly supported the multiple-prototype theory in that they showed that learning a symbol at one size was not equivalent to learning it all sizes. For the same reason, the results rule out scaling as a means to account for uncompensated size change.

The multiple-prototype hypothesis implies that if a well developed prototype set exists, then classification reaction time will not be dependent on the size of the symbols, nor will it depend on learning unfamiliar sizes. These predictions were supported by the results of the second experiment. The classification of alphabetic characters showed no significant relationship between size-familiarity, and classification reaction time.

The third experiment explored the timing relationships within the subprocesses of the classification task. The task may be conceptualized as a two stage process consisting of a preprocessing operation which forms an abstracted representation of the stimulus followed by the actual comparison process. The effect of size change appears to be confined to the preprocessing phase, with an additional 13 milliseconds required following a scale change. Thereafter, the comparison process proceeds in a manner unaffected by the change.

The nature of the additional processing resulting from a scale change is unclear. It may involve resetting some portion of the visual system, selecting a new prototype set or some equally obscure operation. Additional experiments will be necessary to confirm this effect, as well as to uncover the function of the additional processing.

<u>The Revised Kabrisky Model</u>

The Kabrisky model may be easily revised to adequately explain the experimental evidence concerning the effects of scale change. The revised model is illustrated in Fig. 16.

An adaptive scaling "front end", to borrow a term from computer technology, has been added. The input to the system is scaled in a manner that reduces change caused by variations in retinal size due to distance. The scaling factor is controlled by physiological inputs

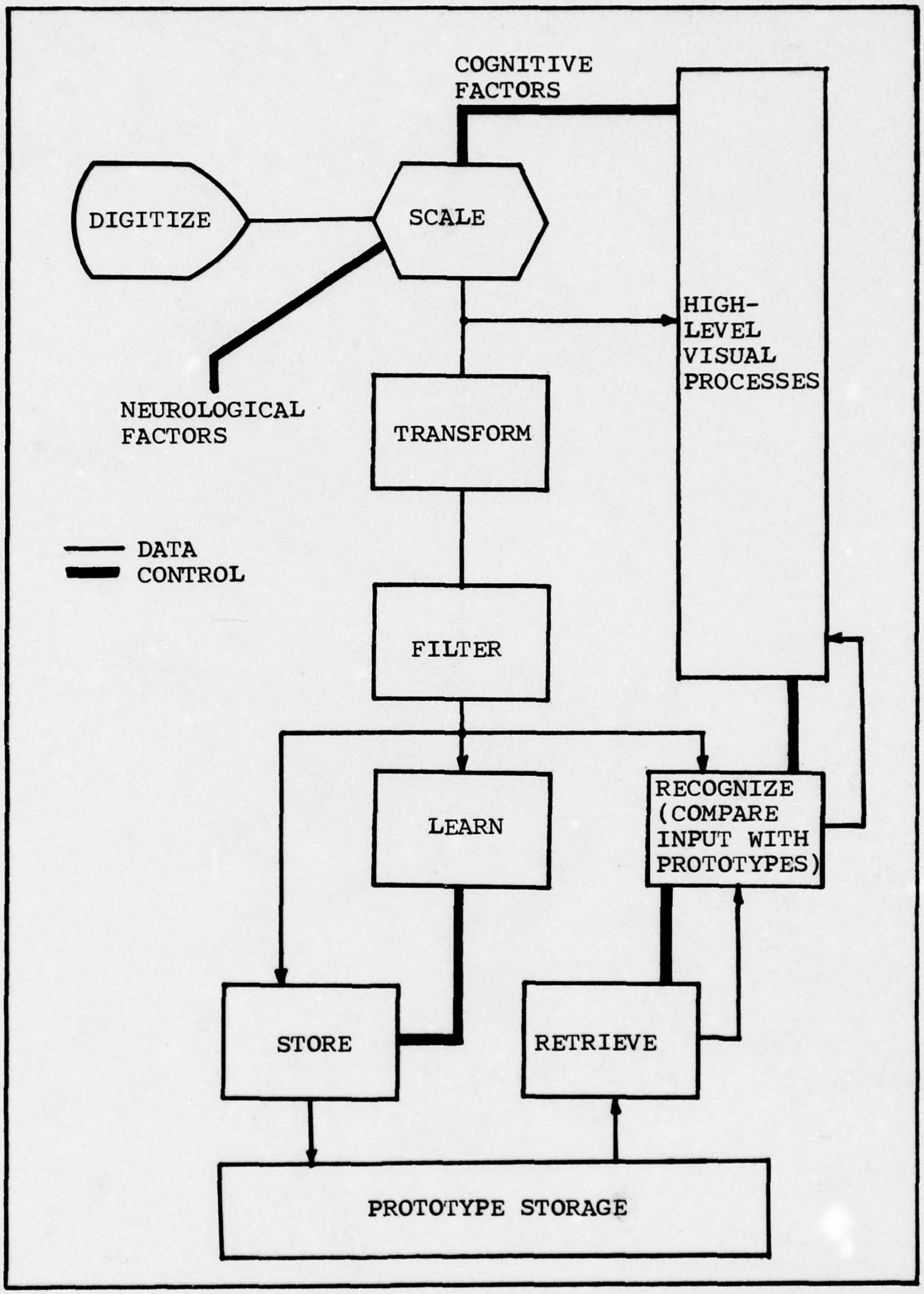

Fig. 16. The Revised Kabrisky Model

as well as by factors derived from scene analysis carried out by higher level cognitive processes.

The process of learning to recognize patterns involves storing prototypes that are not entirely independent of the scale of the stimulus class. The nature of the filtered transform provides for some variance in scale without the need to store additional prototypes. However, when size change exceeds the size represented by the nearest prototype, by perhaps 15%, then an additional prototype is required.

The recognition process remains unchanged from previous formulations of the model. It requires the comparison of stored prototypes with the filtered transform of the input pattern.

Recommendations

The validity of the revised model should be examined through a continuing series of experiments exploring the effect of scale change.

Experiments with larger target sets would serve to test the validity of the third experiment. The range of the linear relationship between target set size and reaction time could be determined.

Several predictions derived from the model may be easily tested. Size changes associated with distance variations should be compensated by scaling, and should not effect reaction time. The relationship between

graduated size change and the requirement for additional prototype storage could be examined using abstract symbol sets.

Perhaps the most intriguing area for study is the nature of the additional processing required after a scale change. Numerous questions await answers. Is the processing time dependent on the magnitude of the scale change? Is it present for only uncompensated scale change, or for change associated with distance variations as well? Attention focused on this area will provide answers that will enable the human visual system model to be further developed and refined.

Bibliography

1. Ankeny, L. A. _The Classification of Chinese Characters by Spatial Filtering_. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1971.

2. Blakemore, C., _et al_. "The Site of Size Constancy." _Perception_, _1_: 111-119 (1972).

3. Bower, T. G. "The Visual World of Infants." in _Perception: Mechanisms and Models, Reading from Scientific American_. San Francisco: W. H. Freeman and Co., 1972.

4. Carl, J. W. _Generalized Harmonic Analysis for Pattern Recognition: A Biologically Derived Model_. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1969.

5. Duser de Barenne, H., _et al_. "Physiological Neuronography of the Cortico-striatal Connections." _Association for Research in Nervous and Mental Disorders 21_: 246-266 (1942).

6. -----. _The Isocortex of the Chimpanzee_. Urbana, Ill.: University of Illinois Press, 1951.

7. Gibson, J. _The Perception of the Visual World_. Cambridge, Mass.: The Riverside Press, 1950.

8. Gill, R. A. _The Scaling Problem in the Classification of Images by Spatial Filtering_. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1970.

9. Ginsburg, A. P. _Psychological Correlates of a Model of the Human Visual System_. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, May 1970.

10. -----. "Psychological Correlates of a Model of the Human Visual System." in _Proceedings 1971 National Aerospace Electronics Conference_. Dayton, Ohio: IEEE, May 1971.

11. Gregory, R. L. The Intelligent Eye. London: Weidenfeld and Nicolson, 1970.

12. Guyton, A. C. Textbook of Medical Physiology (Fourth Edition). Philadelphia: W. B. Saunders Co., 1971.

13. Hall, C. F. The Analysis and Classification of Aperiodic Signals. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1970.

14. Holway, A. H. and Boring, E. G. "Determinants of Apparent Visual Size with Distant Variant." American Journal of Psychology, 54: 21-37 (1941).

15. Huble, D. H. "The Visual Cortex of the Brain." in Perception: Mechanisms and Models, Readings from Scientific American. San Francisco: W. H. Freeman and Co., 1972.

16. Huble, D. H. and Wiesel, T. N. "Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex." Journal of Physiology 195: 215-293 (1968).

17. Kabrisky, Matthew. A Proposed Model for Visual Information Processing in the Human Brain. Urbana, Illinois: University of Illinois Press, 1966.

18. Mahaffey, W. D. Jr. Pattern Recognition Model Based on Cortical Anatomy. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1971.

19. Maher, F. A. "A Correlation of Human and Machine Pattern Discrimination." in Proceedings 1970 National Aerospace Electronics Conference. Dayton, Ohio: IEEE, 1970.

20. Neyman, R. W. Computer Identification of Phonemes in Continuous Speech. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, June 1970.

21. Polyak, S. L. The Retina. Chicago: University of Chicago Press, 1941.

22. Radoy, C. H. Pattern Recognition by Fourier Series Transformations. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1967.

23. Ragsdale, W. O. A Digital Simulation of Psychological Correlates of a Model of the Human Visual System. Unpublished thesis. Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, May 1972.

24. Rock, I. An Introduction to Perception. New York: Macmillan Publishing Co., 1975.

25. Sternberg, S. "High-speed Scanning in Human Memory." in Information Processing Approaches to Visual Perception, edited by R. N. Haber. New York: Holt Rinehart and Winston Inc., 1969.

26. -----. "Two Operations in Character Recognition: Some Evidence from Reaction Time Measurements." from Models for the Perception of Speech and Visual Form, edited by Weiant Wathen-Dunn. Cambridge, Mass: M.I.T. Press, 1967.

27. Tallman, O. H. The Classification of Visual Images by Spatial Filtering. Unpublished dissertation. Wright Patterson Air Force Base, Ohio: Air Force Institute of Technology, March 1967.

28. -----. Processing of Visual Imagery by an Adaptive Model of the Visual System: Its Performance and Its Significance. AMRL TR 70-45. Wright-Patterson Air Force Base, Ohio: Aerospace Medical Research Laboratory, 1970.

29. Thouless, R. H. "Phenomenal Regression to the Real Object." British Journal of Psychology, 21: 339-359, 1931.

Appendix A

VPAF Hardware Schematics

Figure A1 provides a high-level overview of the Display/Response Interface. This interface is the only special purpose logic required in the VPAF system. The interface allows the NOVA-2 to control the display and monitor inputs for the response unit. Figures A2 through A4 provide detailed schematics of the interface logic. Table A-I provides a list of the signals between the interface and the CT-1024 control logic. Table A-II lists the integrated circuits referenced in the schematics.

The interrupt logic consists of four flip-flops and the associated control logic that allows the interface to interrupt the CPU. An interrupt is generated when the Done flip-flop is set. The Set Done signal is generated when a response in received from the response unit or the Write Comp signal is returned from the display terminal. The interrupt logic may be enabled, disabled, or masked out under control of the CPU.

The response logic interfaces with the response unit providing temporary storage for each response. A 4-bit register stores responses. Each response generates a Set Done pulse. The CPU will respond to a response unit

interrupt by issuing a DATIA signal. This signal strobes the contents of the response register onto the data bus.

The character control logic transfers ASCII characters to the terminal. This logic simulates the operation of the terminal keyboard. The ASCII data is placed on six parallel lines; a keyboard strobe (Start Write) pulse is then issued. The strobe transfers the data into the keyboard buffer of the terminal. It also generates a signal indicating the buffer contains a character to be read into the display memory. The terminal will return a Write Comp signal after the character has been transferred to the display memory. The Write Comp generates a Set Done signal, setting the Done flip-flop.

The curser control logic simulates the curser control board. The logic supplies the required negative-going pulses in order to control curser position, page select, and memory clearing.

Several minor modifications to the CT-1024 terminal were required. The keyboard and the curser control board were not used, as their functions were provided by the Display/Response Interface. The Write Comp signal was tapped off of IC11 pin 18. The ENDIS (Enable Display) signal, generated by the interface, was used to enable the display. This required a modification of the video output portion of the terminal. The character information is normally applied to two inputs of an AND gate

(IC17, pins 12, 13) in parallel. The connection between the two pins was broken, and the ENDIS signal was connected to pin 12. The ENDIS signal thus provided an enabling level that allowed character information to pass into the output stream.

Further insight into the operation of the interface logic and the terminal may be gained by the examination of the NOVA-2 Users' Manual and the CT-1024 Technical Manual.

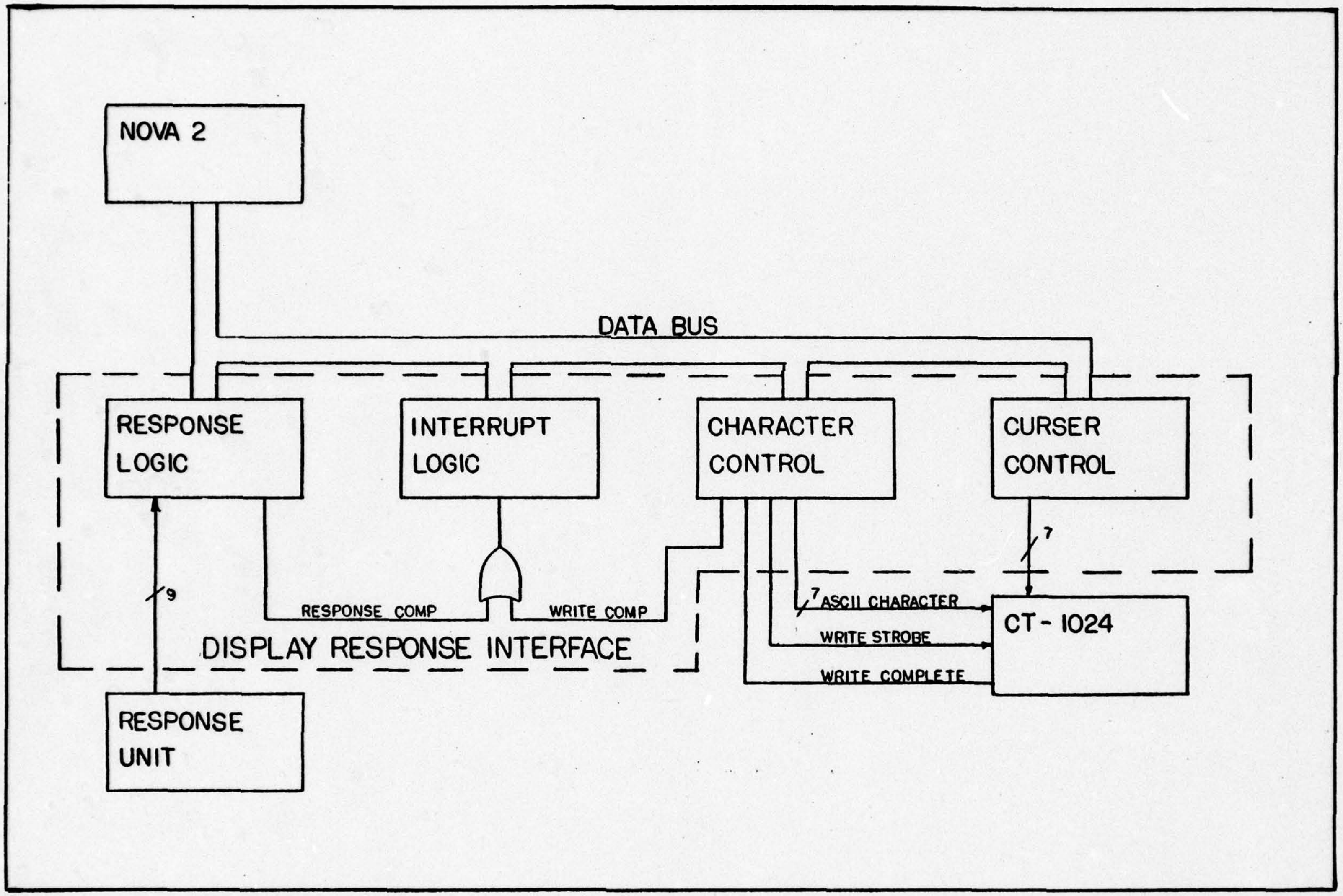

Fig. A1. Display/Response Interface

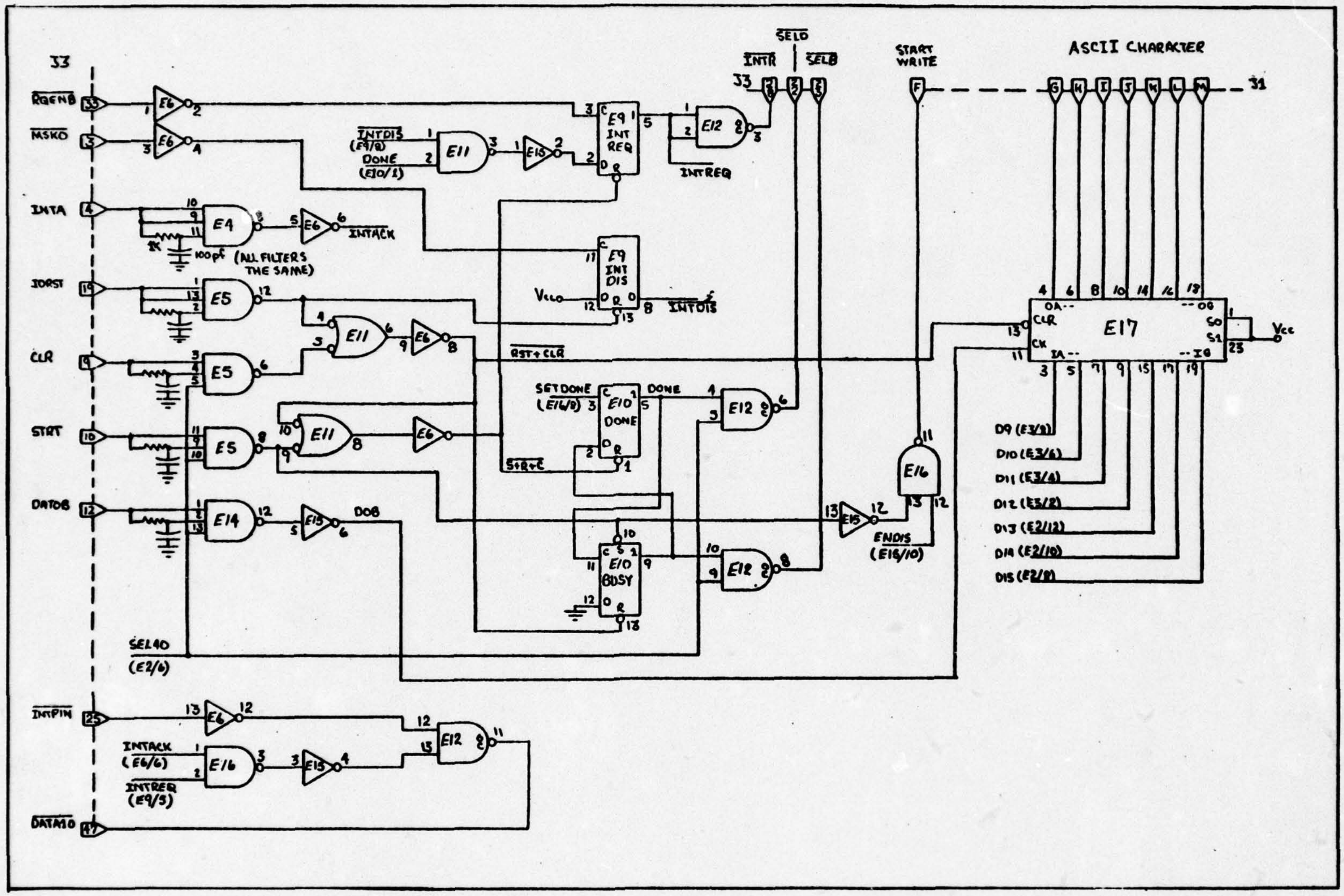

Fig. A2. Interrupt Logic and Control

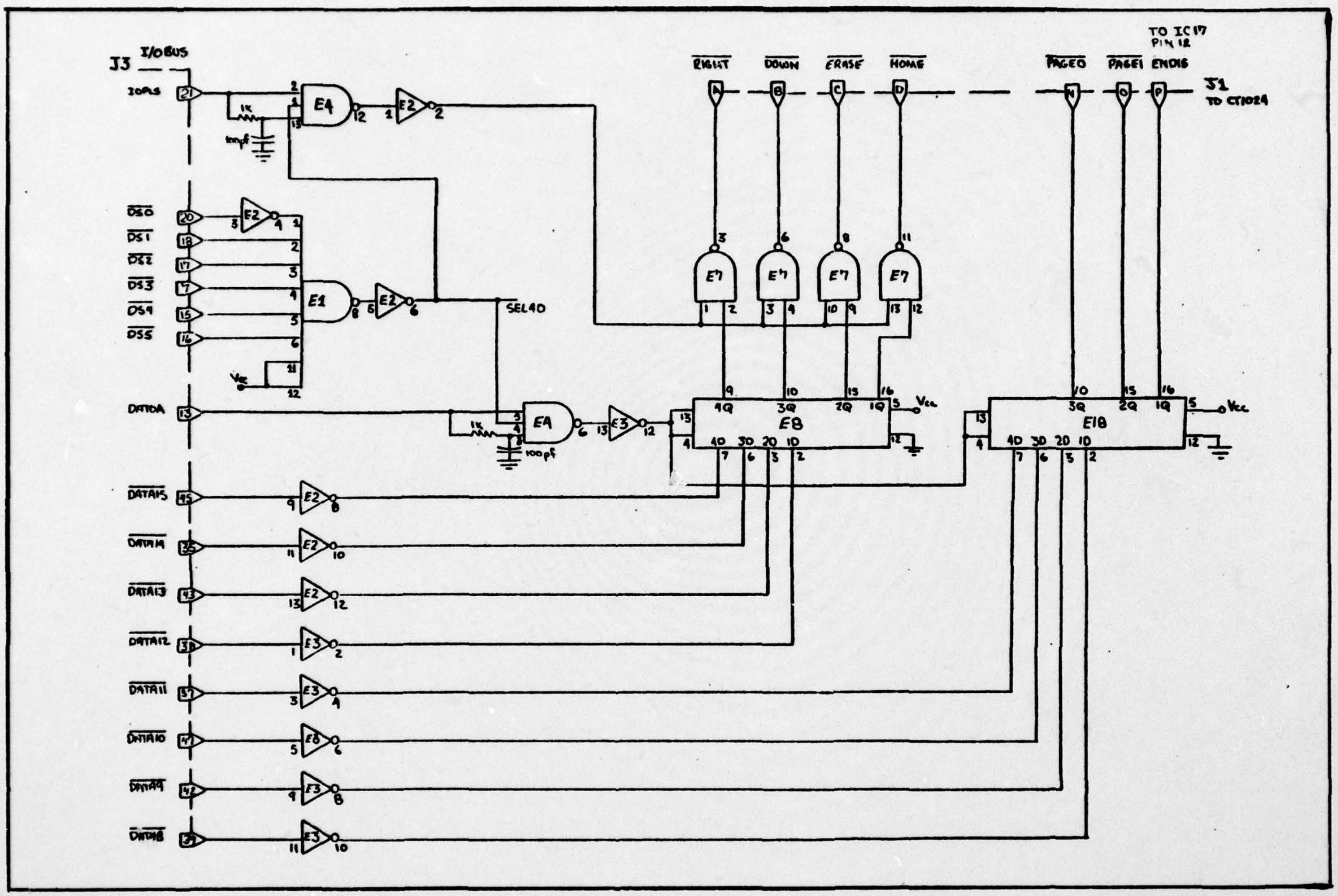

Fig. A3. Curser Control

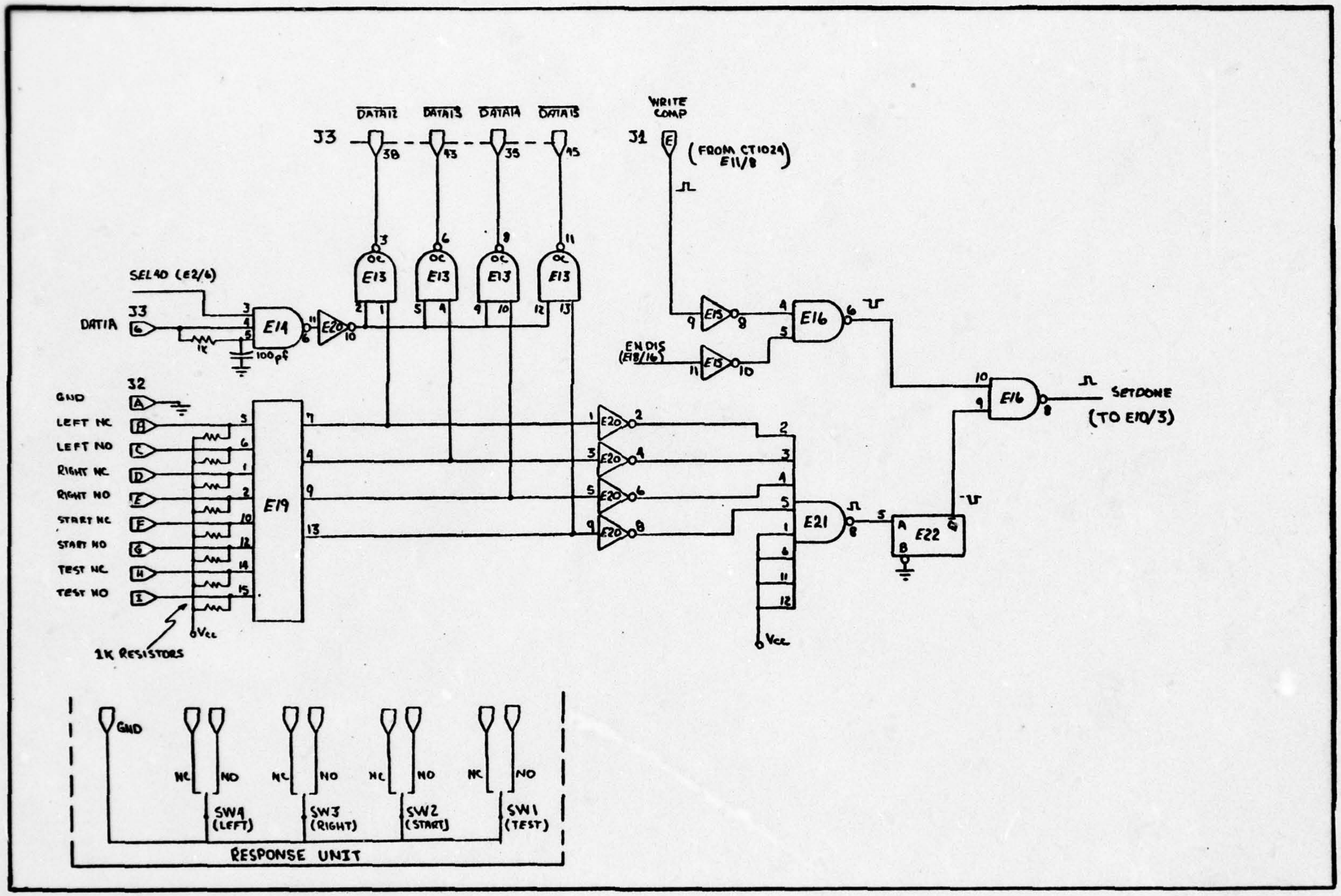

Fig. A4. Response Logic

Table A-I

Interface-Display Terminal Signals

Signal	Interface	Terminal
WRITE COMP	J1/E	E11/8
START WRITE	J1/F	J1/13
ENDIS	J1/P	E17/12
ASCII 0	J1/M	J1/8
ASCII 1	J1/L	J1/9
ASCII 2	J1/K	J1/10
ASCII 3	J1/J	J1/11
ASCII 4	J1/I	J1/12
ASCII 5	J1/H	J1/13
ASCII 6	J1/G	J1/14
$\overline{\text{RIGHT}}$	J1/H	J3/2
$\overline{\text{LEFT}}$	J1/B	J3/3
$\overline{\text{ERASE}}$	J1/C	J3/8
$\overline{\text{HOME}}$	J1/D	J3/1
$\overline{\text{PAGE0}}$	J1/N	J11/1
$\overline{\text{PAGE1}}$	J1/O	J10/2

Table A-II

Interface Component Specifications

Component	Type	V_{cc}	Ground
E1, 21	7430	14	7
E2, 3, 4, 15, 20	7404	14	7
E4, 5, 14	7410	14	7
E7, 11, 16	7400	14	7
E8, 18	7475	12	5
E9, 10	7474	14	7
E17	74198	24	12
E19	74279	16	8
E22	74121	14	7

Appendix B

VPAF Software Description

This appendix contains a brief description of the VPAF software as well as the program listings for the modules utilized for the Experiment One test sequence.

The basic philosophy underlying the design of the software is to concentrate the high-level control of the experiment within one module. Other lower-level modules support the control module by performing data formatting, device control, interrupt processing and other essential real time tasks. The control module executes a wait loop until an interrupt is detected. It then takes the appropriate course of action, depending on the nature of the interrupt.

The resulting software structure is shown below. SYSC is the main control program. The lower-level programs are grouped according to their functional characteristics.

Main Control Level

SYSC	Controls the sequence of the experiment

Second Level Control Functions

STRT	Initializes each trial, starts the real time clock and enables the response interface
CORT	Determines if the response was correct

GDIS	Selects the next display and outputs it
PRNT	Outputs trial results to the line printer
OMES	Outputs feedback message to the display
DLAY	Provides for a variable delay between trials

Data Format

BAD	Converts 16-bit binary words to ASCII characters
TSEQ	Defines the display sequence as well as each individual display

Line Printer Control

WLINE	Outputs an ASCII line of text to the line printer

Display Control

CURS	Controls curser position and erases the display memory
COUT	Outputs a single ASCII character to the display
LOUT	Outputs a line of text to the display
OFF	Turns the display off
ON	Turns the display on
SELO	Selects page zero
SEL1	Selects page one
SOUT	Outputs a display in the form of a 16 by 32 array of ASCII characters

Interrupt Level Control

INTPR	Determines the cause of the interrupt and branches to the appropriate control module
RESIP	Processes response unit interrupts
CLKIP	Processes real time clock interrupts

```
;*********************************************
;
;        SYSTEM CONTROL SEQUENCER
;        CONTROL THE TEST SEQUENCE
;        INITIALIZES CONSTANTS AND FLAGS
;
;*********************************************
        .TITL SYSC
        .ENT     .DLAY
        .ENT  SYSC,DONEF,RSTAT,DADD,TRYN,CORTF,TESTF
        .EXTD TIME,TIME1,.OFF,.ON,.INTP,.LOUT,.SOUT
        .EXTD .CLON,CRATE,HOME,.CURS,EOF,DN
        .EXTD .PRNT,.SEL0,.SEL1,CLONF,OTIMF
        .EXTD .TAB1,.TAB2
        .TXTM 1             ;PACK TEXT FROM LEFT TO RIGHT
;
;       PAGE ZERO POINTERS, FLAGS, CONSTANTS
        .ZREL
RSTAT:  0                   ;RESPONCE UNIT STATUS(L,RT,STRT,TST)
TESTF:  0                   ;=1 IN THE TEST MODE
CORTF:  0                   ;=1 IF RESPONCE WAS CORRECT
DONEF:  0                   ;=1 AFTER SUBJECT RESPONDS
TRYN:   1                   ;CURRENT TRIAL NUMBER
PMAX:   70                  ;MAX PRACTICE TRIAL ALLOWED
TMAX:   32                  ;MAX TEST TRIAL ALLOWED
DADD:   0                   ;POINTER TO ADD. OF CURRENT DISPLAY
.CORT:  CORT                ;ADDRESS OF CORT ENTRY POINT
.OMES:  OMES                ;ADDRESS OF OMES ENTRY POINT
.GDIS:  GDIS                ;ADDRESS OF GDIS ENTRY POINT
.STRT:  STRT                ;ADDRESS OF STRT ENTRY POINT
.DLAY:  DLAY                ;DELAY SUBROUTINE ENTRY POINTER
```

```
        .NREL
SYSC:   SUB     0,0      ;GENERATE A ZERO
        STA     0,TIME   ;RESET TIME
        STA     0,TRYN   ;RESET TRIAL NUMBER
        STA     0,CLONF  ;RESET CLOCK ON FLAG
        STA     0,DONEF  ;RESET DONE FLAG
        STA     0,TESTF  ;RESET THE TEST FLAG
        SUBZL   0,0      ;GENERATE A 1
        STA     0,CORTF  ;INITIALIZE CORRECT FLAG
        LDA     0,.INTP  ;GET INTERRUPT PROCESSOR ENTRY
        STA     0,1      ;STORE IT IN LOC 1
        LDA     2,HOME   ;SET UP TO HOME CURSER
        JSR@    .CURS    ;DO IT
        LDA     2,EOF    ;GET ERASE DISPLAY CONTROL BIT
        JSR@    .CURS    ;ERASE THE DISPLAY
        JSR@    .ON      ;TURN ON DISPLAY, THIS
                         ;WILL PREVENT THE START PULSE
                         ;FROM CAUSING AN INTERRUPT
        IORST            ;ENSURE EVERYTHING IS RESET
        INTEN            ;ENABLE INTERRUPTS
        NIOS    40       ;START THE RESPONCE INTERFACE
;       ENTER THE WAIT LOOP
;       CHECK ON TIME AND DONE FLAG
LOOP3:  LDA     1,CLONF  ;GET CLOCK ON FLAG
        MOV     1,1,SNR  ;SKIP IF CLOCK IS ON
        JMP     TESTD    ;GO TEST THE DONE FLAG
        LDA     1,TIME   ;GET THE CURRENT TIME
        LDA     0,TIME1  ;GET THE OFF TIME
        SUBZ    1,0,SNR  ;SKIP IF TIME NE TIME1
        JSR@    .OFF     ;TURN OF THE DISPLAY
TESTD:  LDA     1,DONEF  ;GET THE DONE FLAG
        MOVR    1,1,SNC  ;IS IT SET?
        JMP     LOOP3    ;NO, SO LOOP
;       DONE INDICATION MEAN SUBJECT HAS RESPONDED
;       PROCESS RESPONCE
        JSR@    .OFF     ;ENSURE DISPLAY IS OFF
        LDA     1,RSTAT  ;GET RESPONCE STATUS
        MOVR    1,1,SZC  ;TEST SEQUENCE REQUEST ?
        JMP     TEST     ;YES, SO PROCESS IT
        MOVR    1,1,SZC  ;START REQUEST ?
        JMP     GO       ;START THE SEQUENCE
;       NOT START OR TEST REQ. SO MUST BE
;       A TRIAL RESPONCE
        LDA     2,HOME   ;GET CONTROL CODE FOR HOME
        JSR@    .CURS    ;SEND CURSER HOME
        LDA     2,EOF    ;SET UP TO ERASE SCREEN
        JSR@    .CURS    ;DO IT
        JSR@    .CORT    ;DETERMINE IF RESP. WAS CORRECT
        ;SET UP 1 SEC DELAY PRIOR TO FEEDBACK MESSAGE
        LDA     2,D100   ;GET COUNT FOR 1 SEC
        JSR@    .DLAY    ;DELAY
        JSR@    .OMES    ;OUTPUT FEEDBACK MESSAGE
        JSR@    .PRNT    ;PRINT TRIAL RESULTS
SSEQ:   JSR@    .GDIS    ;SET UP NEXT DISPLAY
;       GDIS WILL SET THE DONEF FLAG IF THE TEST
;       SEQUENCE IS COMPLETE
```

```
         LDA     1,DONEF ;GET THE DONE FLAG
         MOVR    1,1,SZC ;IS IT SET ?
         JMP     FIN     ;YES, DONE WITH TEST SEQ.
         JSR@    .STRT   ;START THE NEXT TRIAL
         JMP     LOOP3   ;REENTER WAIT LOOP
;        FINISHED WITH TEST SEQ
;        OUTPUT MESSAGE AND RETURN TO START
FIN:     JSR@    .OFF    ;TURN OFF THE DISPLAY
         LDA     2,HOME  ;SET UP TO HOME CURSER
         JSR@    .CURS   ;DO IT
         LDA     2,EOF   ;SET UP TO ERASE SCREEN
         JSR@    .CURS   ;DO IT
         LDA     2,FMES  ;GET MESSAGE POINTER
         JSR@    .LOUT   ;OUTPUT THE MESSAGE
         JSR@    .ON     ;TURN ON THE DISPLAY
         HALT            ;DONE WITH TEST SEQ
;        TEST SEQUENCE SELECTED SET IT UP
TEST:    SUBZL   1,1     ;GENERATE A ONE
         STA     1,CORTF ;INITALIZE CORRECT FLAG
         STA     1,TESTF ;SET THE TEST FLAG
         SUB     1,1     ;GENERATE A ZERO
         STA     1,TRYN  ;INITIALIZE TRIAL NUMBER
         JMP     SSEQ    ;START THE SEQUENCE
;        COME HERE TO RESTART PRACTICE SEQUENCE
GO:      IORST           ;RESET I/O
         SUB     0,0     ;GENERATE A 0
         STA     0,CLONF ;INITIALIZE CLOCK ON FLAG
         STA     0,TIME  ;RESET TIME COUNT
         STA     0,TESTF ;RESET TEST FLAG
         STA     0,TRYN  ;RESET TRIAL NUMBER
         SUBZL   0,0     ;GENERATE A 1
         STA     0,CORTF ;ITITIALIZE CORRECT FLAG
         JMP     SSEQ    ;START NEXT DISPLAY
FMES:    .+1
         .TXT    /TEST COMPLETE, THANK YOU/
D100:    144             ;COUNT FOR 1 SEC DELAY
```

```
;
;
;
;         SUBROUTINE CORT                                          :
;         DETERMINES IF THE TRIAL RESPONCE WAS CORRECT AND
;         SETS THE CORTF FLAG IF IT WAS
;         INPUT: AC1 CONTAINS THE RESPONCE RIGHT
;                SHIFTED 2 BITS
;
CORT:     SUBZL    0,0      ;GENERATE A 1
          STA      0,CORTF  ;INITALIZE CORRECT FLAG TO 1
          LDA@     0,DADD   ;GET FIRST WORD OF DISPLAY
          MOVL     0,0,SZC  ;WAS BIT ZERO A 1 ?
          JMP      TARG     ;YES, THEREFORE IT WAS IN TARGET SET
;         NOT TARGET SET, SO CHECK FOR LEFT RESPONCE
          MOVR     1,1,SZC  ;IS RIGHT RESPONCE BIT=1
          JMP      NGOOD    ;YES, THEREFORE WRONG RESPONCE
          JMP      DONE     ;CORRECT RESPONCE, SO DONE
;TARGET SET SO CHECK FOR RIGHT RESPONCE
TARG:     MOVR     1,1,SZC  ;IS RIGHT RESPONCE BIT=1 ?
          JMP      DONE     ;YES, SO WE ARE DONE
NGOOD:    SUB      1,1      ;GENERATE A 0
          STA      1,CORTF  ;ZERO CORRECT FLAG
DONE:     JMP      0,3      ;RETURN
```

```
;
;
;       SUBROUTINE GDIS
;       GETS DISPLAY SPECIFIED BY TRYN AND OUTPUTS IT
;
;
;
;       SUBROUTINE GDIS
;       GETS DISPLAY SPECIFIED BY TRYN AND OUTPUTS IT
;
GDIS:   STA     3,GRET  ;STORE RETURN ADDRESS
        SUB     1,1     ;GENERATE A 0
        STA     1,DONEF ;RESET THE DONE FLAG
        LDA     1,TESTF ;GET THE TEST FLAG
        MOVR    1,1,SZC ;ARE WE IN TEST MODE ?
        JMP     TMODE   ;YES
;PRACTICE MODE
        ;FIRST CHECK RESPONCE AND INCREMENT TRIAL
        ;NUMBER IF RESPONCE WAS CORRECT
        LDA     1,CORTF ;GET CORRECT FLAG
        MOVR    1,1,SZC ;SKIP IF INCORRECT
        ISZ     TRYN    ;CORRECT, SO INCREMENT TRIAL
        LDA     1,TRYN  ;GET CURRENT TRIAL NO
        LDA     3,PMAX  ;GET MAX PRACTICE NO
        SUBO#   3,1,SZR ;IS TRYN EQ PMAX
        JMP     .+2     ;NO, SO CONTINUE
        SUBZL   1,1     ;GENERATE A ONE TO START OVER
        STA     1,TRYN  ;INITALIZE TRIAL NUMBER
        LDA     3,.TAB1 ;GET POINTER TO PRACT. TABL
        ADD     1,3     ;FORM A POINTER TO DISPLAY ADDRESS
        JMP     CONT    ;CONTINUE
;       TEST MODE
TMODE:  ISZ     TRYN    ;INCREMENT TRIAL NO
        LDA     1,TRYN  ;GET TRIAL NUMBER
        LDA     3,TMAX  ;GET MAX TEST NUMBER
        SUBO#   3,1,SNR ;IS TRYN EQ TMAX
        JMP     FINSH   ;YES, TEST IS COMPLETE
        LDA     3,.TAB2 ;NO,SO CONTINUE WITH THE TEST
        ADD     1,3     ;ADD TO GET ADDRESS POINTER
CONT:   LDA     2,0,3   ;GET DISPLAY ADDRESS FROM TABLE
        STA     2,DADD  ;STORE IN CURRENT DISPLAY ADDRESS
        JSR@    .OFF    ;TURN OFF THE DISPLAY
        JSR@    .SELO   ;SELECT PAGE 0
        JSR@    .SOUT   ;OUTPUT THE DISPLAY
        JMP@    GRET    ;RETURN
FINSH:  SUB     1,1     ;GENERATE A 0 TO START OVER
        STA     1,TRYN
        JMP     TMODE   ;CONTINUE TESTING
        JMP@    GRET    ;RETURN
GRET:   0               ;STORAGE FOR RETURN ADDRESS
```

```
;
;
;        SUBROUTINE OMES
;        OUTPUTS MESSAGES TO THE TEST SUBJECT
;
OMES:    STA      3,ORET   ;STORE RETURN ADDRESS
         ;FIRST POSITION CURSER TO CENTER LINE OF DISPLAY
         LDA      2,DN     ;GET DOWN CONTROL WORD
         JSR@     .CURS    ;MOVE IT DONE ONE LINE
         LDA      2,DN     ;ETC
         JSR@     .CURS
         LDA      2,DN
         JSR@     .CURS
         LDA      0,CORTF  ;GET THE CORRECT FLAG
         MOV      0,0,SZR  ;WAS RESPONCE CORRECT ?
         JMP      YES      ;YES, SO OUTPUT GOOD MESSAGE
NO:      LDA      2,BADM   ;GET POINTER TO BAD MESSAGE
         JMP      OUT      ;OUTPUT IT
YES:     LDA      2,.GOOD  ;GET GOOD MESSAGE
OUT:     JSR@     .LOUT    ;OUTPUT THE MESSAGE
         JSR@     .ON      ;DISPLAY MESSAGE
         ;NOW SET UP MESSAGE DISPLAY DELAY
         JMP@     ORET             ;RETURN
ORET:    0                 ;STORAGE FOR RETURN ADDRESS
.GOOD:   .+1               ;POINTER TO GOOD MESSAGE
         .TXT     /                CORRECT/
BADM:    .+1               ;POINTER TO BAD MESSAGE
         .TXT     /               WRONG/
```

```
;
;
;        SUBROUTINE STRT
;        START THE TRIAL SEQUENCE
;         STARTS THE CLOCK AND
;         THE RESPONCE INTERFACE
;
STRT:    STA      3,SRET   ;SAVE RETURN ADDRESS
         LDA      3,OTIME  ;GET ON TIME EVENT COUNT
         STA      3,TIMEI  ;STORE IN TIME COMPARE LOC
         JSR@     .ON      ;TURN TO THE DISPLAY
         JSR@     .CLON    ;TURN ON THE CLOCK
         NIOS     40       ;RESTART THE INTERFACE
         INTEN             ;ENABLE INTERRUPTS
         JMP@     SRET     ;RETURN
SRET:    0                 ;STORAGE FOR RETURN ADDRESS
```

```
;
;
;
;        SUBROUTINE DLAY
;        DELAYS THE NUMBER OF ONE HUNDRETHS
;        SECS SPECIFIED BY AC2
;
DLAY:    STA     2,C1      ;STORE HUNDRETHS COUNT
         ISZ     C1        ;ADD 1 TO C1
DC1:     DSZ     C1        ;DECREMENT MAIN LOOP COUNTER
         JMP     .+2       ;NOT DONE YET
         JMP     0,3       ;ITS ZERO SO WE ARE DONE
         LDA     2,HSEC    ;INITIALIZE 1/100 COUNT
         STA     2,C2
         DSZ     C2        ;LOOP FOR 1/100 SEC
         JMP     .-1
         JMP     DC1       ;CHECK MAIN LOOP COUNT AGAIN
C1:      0                 ;MAIN LOOP COUNTER
C2:      0                 ;1/100 SEC COUNTER
HSEC:    5000              ;COUNT FOR 1/100 SEC
         .END              ;END OF CONTROL SUBROUTINES
```

```
;************************************
;
;        TEST SEQUENCE 1
;        DISPLAY DESCRIPTIONS                    :
;
;        EACH 16 BY 32 DISPLAY IS DESCRIBED BY A 2 BY 16
;        WORD ARRAY.  A BIT SET IN THE ARRAY CAUSES A
;        CHARACTER TO BE PRINTED IN THE CORRESPONDING
;        POSITION IN THE DISPLAY.
;
;************************************
         .TITL   TSEQ3
         .ENT    .TAB1,.TAB2
        ..ZREL
         ;SET UP ENTRY POINTS TO DISPLAY LISTS
.TAB1:   TAB1     ;POINTER TO PRACTICE TABLE
.TAB2:   TAB2     ;POINTER TO TEST TABLE
         .NREL
         .RDX     2        ;ALL NUMBERS IN BASE 2
;        SET UP DISPLAY SEQUENCES
TAB1:    .        ;STARTING ADDRESS OF TABLE
         S5
         S0
         L6
         S5
         L7
         S0
         L9
         L2
         S4
         L7
         S1
         L6
         S5
         L2
         S8
         L9
         S4
         S1
         S5
         L7
         S5
         S0
         L9
         L6
         S8
         L2
         S1
         L3
         L9
         S4
         S0
         S5
         L2
         L7
         L6
```

AD-A035 290 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 6/4
THE SCALING PROBLEM IN VISUAL PATTERN RECOGNITION.(U)
DEC 76 D D CARPENTER

UNCLASSIFIED GE/EE/76D-17 NL

2 OF 2
AD
A035290

END
DATE
FILMED
3-77

```
        L2
        S8
        L9
        S8
        S1
        S5
        L9
        L3
        S4
        L2
        S8
        L6
        S5
        L2
        S1
        L2
        S4
        L9
        L3
        L9
;
;
;
TAB2:                   ;TEST TABLE POINTER
        S8
        L3
        S0
        S4
        L6
        L1
        L8
        S2
        S8
        L0
        S4
        S7
        L3
        L5
        S1
        S9
        L2
        L4
        L7
        S3
        S0
        L3
        S1
        S6
        L2
```

```
;       DEFINE THE DISPLAYS
;       MAKE FIRST THREE SYMBOLS PART OF
;       THE TARGET SET BY SETTING BIT ONE
;       OF THE DISPLAY EQUAL TO 1
S0:     0000000000000000
         .BLK     1011
        0000000000000001
        0000000000000000
        0000000000000010
        0000000000000000
        0000000000000010
        1100000000000000
        0000000000000001
        0000000000000000
        .BLK     1100
S1:     1111111111111111              ;TARGET SYMBOL
        .BLK     1011
        0000000000000000
        1000000000000000
        0000000000000000
        1100000000000000
        0000000000000001
        0000000000000000
        0000000000000010
        0000000000000000
        .BLK     1100
S2:     1111111111111110              ;TARGET SYMBOL
        .BLK     1011
        0000000000000010
        0000000000000000
        0000000000000001
        0000000000000000
        0000000000000000
        1100000000000000
        0000000000000011
        1000000000000000
        .BLK     1100
S3      0000000000000011
        .BLK     1011
        0000000000000000
        1000000000000000
        0000000000000011
        1100000000000000
        0000000000000000
        0100000000000000
        0000000000000000
        1000000000000000
        .BLK     1100
```

```
S4:     0000000000000100
        .BLK    1011
        0000000000000011
        0000000000000000
        0000000000000000
        1100000000000000
        0000000000000000
        0100000000000000
        0000000000000000
        1000000000000000
        .BLK    1100
S5:     0000000000000101
        .BLK    1011
        0000000000000000
        0100000000000000
        0000000000000000
        1000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        .BLK    1100
S6:     0000000000000110
        .BLK    1011
        0000000000000011
        0000000000000000
        0000000000000001
        0000000000000000
        0000000000000001
        0100000000000000
        0000000000000001
        1000000000000000
        .BLK    1100
S7:     1111111111111001          ;TARGET SYMBOL
        .BLK    1011
        0000000000000000
        1000000000000000
        0000000000000011
        1100000000000000
        0000000000000000
        1000000000000000
        0000000000000000
        1000000000000000
        .BLK    1100
S8:     0000000000001000
        .BLK    1011
        0000000000000011
        0000000000000000
        0000000000000001
        1100000000000000
        0000000000000001
        0000000000000000
        0000000000000001
        1100000000000000
        .BLK    1100
```

```
S9:     1111111111110111        ;TARGET SYMBOL
        .BLK    1011
        0000000000000001
        0000000000000000
        0000000000000001
        0000000000000000
        0000000000000010
        0000000000000000
        0000000000000001
        1100000000000000
        .BLK    1100
;       DEFINE THE TIMES TWO SIZE DISPLAYS
LO:     0000000001100100
        .BLK    111
        0000000000000011
        0
        0000000000000011
        0
        0000000000001100
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000001100
        1111000000000000
        0000000000001100
        1111000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        .BLK    1000
L1:     1111111110011011        ;TAGRET SET
        .BLK    111
        0000000000000000
        1100000000000000
        0000000000000000
        1100000000000000
        0000000000000000
        1111000000000000
        0000000000000000
        1111000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000001100
        0000000000000000
        .BLK    1000
```

```
L2:     1111111110011010        ;TARGET SET
        .BLK    111
        0000000000001100
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000000
        1111000000000000
        0000000000000000
        1111000000000000
        0000000000001111
        1100000000000000
        0000000000001111
        1100000000000000
        .BLK    1000
L3:     0000000001100111
        .BLK    111
        0000000000000000
        1100000000000000
        0000000000000000
        1100000000000000
        0000000000001111
        1111000000000000
        0000000000001111
        1111000000000000
        0000000000000000
        0011000000000000
        0
        0011000000000000
        0
        1100000000000000
        0
        1100000000000000
        .BLK    10000
L4:     0000000001101000
        .BLK    111
        0000000000001111
        0
        0000000000001111
        0
        0000000000000000
        1111000000000000
        0000000000000000
        1111000000000000
        0
        0011000000000000
        0
        0011000000000000
        0000000000000000
        1100000000000000
        0000000000000000
        1100000000000000
        .BLK    1000
```

```
L5:     0000000001101001
        .BLK    111
        0000000000000000
        0011000000000000
        0000000000000000
        0011000000000000
        0000000000000000
        1100000000000000
        0000000000000000
        1100000000000000
        0000000000001111
        0000000000000000
        0000000000001111
        0000000000000000
        0000000000001111
        0000000000000000
        0000000000001111
        0000000000000000
        .BLK    1000
L6:     0000000001101010
        .BLK    111
        0000000000001111
        0000000000000000
        0000000000001111
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        1100000000000000
        0000000000000011
        1100000000000000
        .BLK    1000
L7:     1111111110010101            ;TARGET SET
        .BLK    111
        0000000000000000
        1100000000000000
        0000000000000000
        1100000000000000
        0000000000001111
        1111000000000000
        0000000000001111
        1111000000000000
        0000000000000000
        1100000000000000
        0000000000000000
        1100000000000000
        0
        1100000000000000
        0
        1100000000000000
        .BLK    1000
```

```
L8:     0000000001101100
        .BLK    111
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        1111000000000000
        .BLK    1000
L9:     1111111110010011        ;TARGET SET
        .BLK    111
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        1111000000000000
        .BLK    1000
        .END                    ;END OF DISPLAY DEFINITIONS
```

```
L8:     0000000001101100
        .BLK    111
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        1111000000000000
        .BLK    1000
L9:     1111111110010011            ;TARGET SET
        .BLK    111
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000000011
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000001100
        0000000000000000
        0000000000000011
        1111000000000000
        0000000000000011
        1111000000000000
        .BLK    1000
        .END                ;END OF DISPLAY DEFINITIONS
```

```
;*********************************************************
;
;
;                                                              :
;          DISPLAY CONTROL SUBROUTINES
;
;
;************************************************************
;
;          PAGE ZERO POINTERS AND CONSTANTS
;
           .TITL    DISUB
           .ENT .SELO,.SEL1,.ON,.OFF,.CURS,.SOUT,.LOUT
           .ENT M200,M377,HOME,EOF,DN,RT
           .ENT M40,M100,CHAR
           .ZREL
;          SUBROUTINE POINTERS
.SELO:     SELO
.SEL1:     SEL1
.ON:       ON
.OFF:      OFF
.CURS:     CURS
.LOUT:     LOUT
.SOUT:     SOUT
.COUT:     COUT
;
;          CONSTANTS
M40:       40
M100:      100
M200:      200
HOME:      10                   ;COMMAND CODE FOR HOME
EOF:       4                    ;COMMAND CODE FOR ERASE SCREEN
DN:        2                    ;COMMAND CODE FOR CURSER DOWN
RT:        1                    ;COMMAND CODE FOR CURSER RIGHT
M377:      377                  ;USED TO MASK OUT LEFT BYTE
;
;
;          DISPLAY CONTROL
DSTAT:     0                    ;DISPLAY STATUS WORD
CHAR:      43                   ;CHARACTER FOR DISPLAY OUTPUT
;
;          STORAGE FOR ACCUMULATORS DURING SUBROUTINE
;          PROCESSING
AC0:       0
AC1:       0
AC2:       0
AC3:       0
```

```
        .NREL           ;BEGINING OF NORMAL
                        ;RELOCATABLE CODE
;
;
;       SUBROUTINE SELO
;       SELECTS PAGE ZERO FOR DISPLAY
SELO:   STA     0,AC0               ;STORE ACCUMULATORS
        STA     1,AC1
        LDA     0,DSTAT             ;GET DISPLAY STATUS
        LDA     1,M200              ;GET MASK
        AND     1,0                 ;MASK OUT PAGE BITS
        LDA     1,M100              ;GET PAGE 0 BIT
        ADD     1,0                 ;ADD IN THE BIT
        STA     0,DSTAT             ;STORE NEW STATUS
        DOA     0,40                ;OUTPUT NEW STATUS
        LDA     0,AC0
        LDA     1,AC1
        JMP     0,3                 ;RETURN
;
;
;       SUBROUTINE SEL1
;       SELECTS PAGE 1 FOR DISPLAY
SEL1:   STA     0,AC0               ;STORE ACCUMULATORS
        STA     1,AC1
        LDA     0,DSTAT             ;GET DEVICE DTATUS
        LDA     1,M200              ;GET MASK BIT
        AND     1,0                 ;WIPE OUT PAGE BIT
        LDA     1,M40               ;GET PAGE 1 BIT
        ADD     1,0                 ;ADD IN NEW PAGE BIT
        DOA     0,40                ;OUTPUT NEW STATUS
        STA     0,DSTAT             ;STORE NEW STATUS
        LDA     0,AC0               ;RESTORE ACCUMULATORS
        LDA     1,AC1
        JMP     0,3                 ;RETURN
;
;
;       SUBROUTINE ON
;       ENABLES CHARACTER OUTPUT TO THE DISPLAY
ON:     STA     0,AC0               ;STORE ACCUMULATORS
        STA     1,AC1
        LDA     0,M200              ;GET ON BIT
        LDA     1,DSTAT             ;GET DEVICE STATUS
        COM     0,0                 ;NOW OR THEM
        AND     0,1
        ADC     0,1
        DOA     1,40                ;OUTPUT NEW STATUS
        STA     1,DSTAT             ;STORE NEW STATUS
        LDA     0,AC0               ;RESTORE ACCUMULATORS
        LDA     1,AC1
        JMP     0,3                 ;RETURN
```

```
;
;
;         SUBROUTINE OFF
;         DISABLES CHARACTER OUTPUT TO THE DISPLAY
OFF:      STA      0,AC0              ;STORE ACCUMULATORS
          STA      1,AC1
          LDA      0,M200             ;GET THE ON BIT
          COM      0,0                ;COMPLIMENT IT
          LDA      1,DSTAT            ;GET DEVICE STATUS
          AND      0,1                ;CLEAR BIT ONE
          DOA      1,40               ;OUTPUT STATUS
          STA      1,DSTAT            ;STORE NEW STATUS
          LDA      0,AC0              ;RESTORE ACCUMULATORS
          LDA      1,AC1
          JMP      0,3                ;RETURN
;
;
;
;         SUBROUTINE COUT
;         OUTPUTS THE CHARACTER CONTAINED IN AC2
;         TO THE DISPLAY
;
COUT:     DOBS     2,40     ;OUTPUT THE CHARACTER
          SKPDN    40       ;WAIT TILL DONE
          JMP      .-1
          NIOC     40       ;CLEAR THE INTERRUPT REQ
          JMP      0,3      ;RETURN
;
;
;         SUBROUTINE CURS
;         OUTPUTS CURSER CONTROL INFORMATION TO THE
;         DISPLAY
;INPUT:   COMMAND CODE IN AC2
;         1=MOVE RIGHT
;         2=MOVE DOWN
;         4=ERASE DISPLAY
;         8=MOVE HOME
;
CURS:     STA      1,AC1    ;STORE AC1
          LDA      1,DSTAT  ;GET CURRENT DISPLAY STATUS
          COM      2,2      ;OR IN COMMAND CODE
          AND      2,1
          ADC      2,1
          DOA      1,40     ;OUTPUT NEW STATUS
          NIOP     40       ;STROBE IT TO THE DISPLAY
          LDA      1,AC1    ;RESTORE AC1
          JMP      0,3      ;RETURN
```

```
;
;
;        SUBROUTINE SOUT
;        OUTPUTS A FULL CRT DISPLAY
;INPUT:  2 BY 16 ARRAY POINTED TO BY AC2
;        CHAR: A PAGE 0 ASCII CHARACTER TO BE DISPLAYED
;
;OUTPUT: EACH ONE BIT IN THE INPUT ARRAY REPRESENTS A POINT
;        IN THE 16 BY 32 DISPLAY GRID WHERE THE SPECIFIED
;        CHARACTER IS TO BE PRINTED. ALL OTHER LOCATIONS
;        WILL BE FILLED WITH BLANKS
;
SOUT:    INTDS              ;ENSURE INTERRUPTS ARE DISABLED
         STA      0,SAC0    ;SAVE AC'S
         STA      1,SAC1
         STA      3,SAC3
         STA      2,POINT   ;SET UP POINTER TO INPUT ARRAY
         LDA      2,HOME    ;GET HOME COMMAND WORD
         JSR@     .CURS     ;SEND THE CURSER HOME
         LDA      2,M40     ;LOAD THE WORD COUNTER
         STA      2,COUNT
LOOP1:   LDA      1,DM16    ;-16 GOES TO SHIFT COUNTER
         LDA@     0,POINT   ;GET WORD FROM ARRAY
         ISZ      POINT     ;SET POINTER TO NEXT WORD
LOOP2:   LDA      2,BL      ;SET UP TO OUTPUT A BLANK
         MOVL     0,0,SZC   ;IS BIT 0 A ONE
         LDA      2,CHAR    ;YES,SET UP CHAR OUTPUT
         JSR@     .COUT     ;OUTPUT CHAR IN AC2
         INC      1,1,SZR   ;16 SHIFTS DONE ?
         JMP      LOOP2     ;NO, SO GET NEXT BIT
         DSZ      COUNT     ;32 WORDS DONE ?
         JMP      LOOP1     ;NO,SO GET NEXT WORD
         LDA      0,SAC0    ; YES,RESTORE AC'S
         LDA      1,SAC1
         JMP@     SAC3      ;RETURN
SAC0:    0                  ;STORAGE FOR AC0
SAC1:    0                  ;STORAGE FOR AC1
SAC3:    0                  ;STORAGE FOR AC3
POINT:   0                  ;POINTER TO INPUT ARRAY
DM16:    -20                ;-16 FOR SHIFT COUNT
BL:      40                 ;ASCII CODE FOR BLANK
COUNT:   0                  ;WORD COUNTER
```

```
;
;         SUBROUTINE LOUT
;         OUTPUTS A LINE TO THE DISPLAY
;         FIRST WORD OF THE LINE IS POINTED TO BY AC2 :
;         END OF LINE IS SPECIFIED BY NULL,CR,
;         OR 32 CHARACTERS
;
LOUT:     INTDS               ;DISABLE INTERRUPTS
          STA       0,LAC0    ;STORE ACCUMULATORS
          STA       1,LAC1
          STA       2,LAC2
          STA       3,LAC3
          LDA       0,M40     ;LOAD CHARACTER COUNTER
          NEG       0,0       ;MAKE IT -32
;         NOW OUTPUT WORD, FIRST LEFT BYTE THEN RIGHT
LOOP:     LDA       3,M377    ;LEFT BYTE MASK TO AC3
          LDA@      1,LAC2    ;GET A WORD FROM THE INPUT LINE
          MOVS      1,2       ;GET LEFT BYTE INTO RIGHT HALF OF AC2
          AND       3,1       ;SAVE RIGHT BYTE IN AC1
          AND       3,2,SNR   ;IS LEFT BYTE A NULL
          JMP       NULL      ;IT WAS,SO EXIT
OUTC:     DOBS      2,40      ;CHARACTER,SO OUTPUT IT
          SKPDN     40        ;WAIT TILL DONE
          JMP       .-1
          LDA       3,CR      ;GET CODE FOR CARRIGE RET
          SUB#      2,3,SNR   ;WAS IT A CR
          JMP       DONE      ;YES, SO WE ARE DONE
          MOV       1,1,SNR   ;CHECK 2'ND BYTE FOR NULL
          JMP       NULL      ;IT WAS
          DOBS      1,40      ;GOOD CHAR SO OUTPUT IT
          SKPDN     40        ;WAIT TILL DONE
          JMP       .-1
          SUB#      1,3,SNR   ;CHECK FOR A CR
          JMP       DONE      ;IT WAS, SO DONE
          INC       0,0,SZR   ;32 CHARS OUT YET ?
          JMP       NEXT      ;NO SO CONTINUE
DONE:     LDA       0,LAC0    ;RESTORE ACCUMULATORS
          LDA       1,LAC1
          JMP@      LAC3      ;RETURN
NULL:     LDA       2,CR      ;NULL DETECTED, OUTPUT CR
          JMP       OUTC
NEXT:     ISZ       LAC2      ;INCREMENT POINTER TO NEXT
                              ;WORD IN INPUT LINE
          JMP       LOOP      ;CONTINUE
CR:       15                  ;CODE FOR CARRIAGE RETURN
LAC0:     0                   ;STORAGE FOR ACCUMULATORS
LAC1:     0
LAC2:     0
LAC3:     0
          .END                ;END OF DISPLAY SUBROUTINES
```

```
;****************************************
;
;       SUBROUTINE WLINE
;       OUTPUTS LINES TO THE LPT                :
;       INPUT:  AC2=POINTER TO OUTPUT ARRAY
;       ROUTINE WILL OUTPUT 80 CHAR MAX
;       IT WILL ALSO TERMINATE ON A NULL OR CR
;       CALLING SEQUENCE:       JSR@ .WLIN
;       WHERE .WLIN IS EXTD
;
;****************************************
        .TITL   WLINE
        .ENT    .WLIN,WLINE
        .ZREL
.WLIN:  WLINE   ;PAGE ZERO ENTRY
        .NREL
WLINE:  STA     0,PAC0  ;STORE AC'S
        STA     1,PAC1
        STA     2,POINT ;STORE INPUT ARRAY POINTER
        STA     3,PRET  ;STORE RETURN
        LDA     3,D40   ;GET MAX WORD COUNT
        STA     3,COUNT ;STORE IT
;       OUTPUT LOOP-FIRST OUTPUT LEFT BYTE
;       OF INPUT WORD THEN RIGHT BYTE, THEN GET
;       THE NEXT WORD
LOOP:   LDA@    1,POINT ;GET THE INPUT WORD
        MOVS    1,2     ;EXCHANGE BYTES
        LDA     0,M377  ;GET LOW ORDER BYTE MASK
;       NOW AND LEAVING LEFT BYTE IN LOW ORDER HALF
;       OF AC2, AND CHECK FOR NULL
        AND     0,2,SNR ;SKIP IF GOOD CHARACTER
        JMP     DONE    ;BYTE WAS A NULL
;       NOW SEE IF BYTE IS A CR
        LDA     3,CR    ;GET ASCII CR
        SUB#    2,3,SNR ;SKIP IF NOT CR
        JMP     DONE    ;CR THEREFORE DONE
;       NOW OUTPUT LEFT BYTE
        SKPBZ   17      ;MAKE SURE LPT ISN'T BUSY
        JMP     .-1
        DOAS    2,17    ;OUTPUT THE CHARACTER
;       NOW SET UP TO OUTPUT THE RIGHT BYTE OF THE
;       INPUT WORD
        MOV     1,2     ;MOVE ORIGINAL WORD TO AC2 .
        LDA     0,M377  ;GET LOW ORDER BYTE MASK
        AND     0,2,SNR ;AND OUT HIGH ORDER BYTE AND
                        ;CHECK FOR NULL
        JMP     DONE    ;NULL,THEREFORE DONE
        LDA     3,CR    ;GET AN ASCII CR
        SUB#    2,3,SNR ;SKIP IF NOT CR
        JMP     DONE    ;CR, THEREFORE DONE
;       NOW OUTPUT THE RIGHT BYTE OF THE INPUT WORD
        SKPBZ   17      ;WAIT IF LPT IS BUSY
        JMP     .-1
        DOAS    2,17    ;OUTPUT IT
        ISZ     POINT   ;INCREMENT TO NEXT INPUT WORD
        DSZ     COUNT   ;CHECK FOR 40 WORDS MAX
```

```
        JMP     LOOP     ;GET THE NEXT WORD
;       COME HERE WHEN DONE
DONE:   LDA     2,CR     ;SET UP TO OUTPUT CR
        SKPBZ   LPT      ;IS LPT BUSY
        JMP     .-1      ;YES SO WAIT
        DOAS    2,LPT    ;OUTPUT CR
        LDA     2,LF     ;SET UP TO OUTPUT LF
        SKPBZ   LPT      ;IS LPT BUSY
        JMP     .-1      ;YES, SO WAIT
        DOAS    2,LPT    ;OUTPUT LF
        SKPDN   LPT      ;WAIT FOR DONE
        JMP     .-1
        NIOC    LPT      ;CLEAR DONE
        LDA     0,PAC0   ;RESTORE AC'S
        LDA     1,PAC1
        JMP@    PRET     ;RETURN
CR:     15
D40:    50               ;DECIMAL 40
PAC0:   0                ;STORAGE FOR AC'S
PAC1:   0
PRET:   0
POINT:  0                ;INPUT ARRAY POINTER
COUNT:  0                ;MAX LOOP COUNTER
LF:     12               ;LINE FEED FOR FINAL OUTPUT
M377:   377              ;BYTE MASK
        .END
```

```
\;*******************************************
;
;         SUBROUTINE BAD
;         CONVERTS BINARY NUMBER TO AN ASCII            :
;         DECIMAL CHARACTER STRING
;         INPUT:   AC1-NUMBER TO BE CONVERTED
;                  AC2 POINTER TO 4 WORD ARRAY TO CONTAIN
;                  OUTPUT CHARACTER STRING
;         OUTPUT:  ASCII STRING TERMINATED BY A NULL
;                  REPRESENTING THE INPUT NUMBER
;         CALLING SEQUENCE:  JSR@ .BAD
;
;*******************************************
          .TITL    BAD
          .ENT     .BAD,BAD
          .ZREL
.BAD:     BAD                ;PAGE ZERO POINTER TO ENTRY
;
          .NREL
BAD:      STA      3,BAD3    ;SAVE RETURN
          STA      2,BAD2    ;SAVE AC2
          STA      1,BAD1    ;SAVE AC1
          STA      0,BAD0    ;SAVE AC0
          LDA      3,TENA    ;ADDRESS OF POWER OF 10 TAB
          STA      3,POINT   ;INITIALIZE POINTER
          LDA      0,MIN     ;ASSUME NEGATIVE
          MOVL#    1,1,SZC   ;SKIP IF +
          NEG      1,1,SKP   ;NEG,MAKE IT POSITIVE
          LDA      0,POS     ;LOAD ASCII +
L1:       STA      1,COUNT   ;SAVE COUNT
          JSR      BUILD     ;PUT CHAR IN OUTPUT ARRAY
          LDA      1,COUNT   ;GET CURRENT VALUE OF COUNT
          LDA@     3,POINT   ;GET CURRENT POWER OF 10
          ISZ      POINT     ;INCREMENT TO NEXT POWER OF 10
          MOV      3,0,SNR
          JMP      NULL      ;OUTPUT NULL
          LDA      0,ZERO    ;GET ASCII ZERO
L2:       SUBZ     3,1,SZC   ;DOES POWER OF 10 GO IN
          INC      0,0,SKP   ;YES,BUMP RESULT DIGIT
          ADD      3,1,SKP   ;NO,RESTORE PREVIOUS VALUE
          JMP      L2        ;CONTINUE SUBTRACTING
          JMP      L1        ;OUTPUT COMPLETED CHARACTER
NULL:     STA@     1,BAD2    ;INSERT NULL WORD
          LDA      0,BAD0    ;RESTORE AC'S
          LDA      1,BAD1
          JMP@     BAD3      ;RETURN
BUILD:    LDA      2,BAD2    ;GET OUTPUT ARRAY POINTER
          MOVL#    2,2,SZC   ;SKIP IF POINTER WAS POSITIVE
          JMP      RBYTE     ;NEG,MEANS STORE RIGHT BYTE
LBYTE:    MOVS     0,0       ;SWAP TO GET CHAR IN LEFT BYTE
          STA@     0,BAD2    ;STORE WORD IN ARRAY
          NEG      2,2       ;NEGATE POINTER
          STA      2,BAD2    ;STORE NEGATIVE OOINTER
          JMP      0,3       ;RETURN
```

```
RBYTE:  NEG     2,2         ;MAKE POINTER POSITIVE
        LDA     1,0,2       ;GET WORD FROM ARRAY
        ADD     0,1         ;ADD IN ASCII CHARACTER
        STA     1,0,2       ;RETURN OUTPUT WORD         :
        STA     2,BAD2      ;RETURN POINTER
        ISZ     BAD2        ;INCREMENT TO NEXT OUTPUT WORD
        JMP     0,3         ;RETURN
;       STORAGE
BAD0:   0           ;STORAGE FOR AC'S
BAD1:   0
BAD2:   0
BAD3:   0
        .RDX 10
TENT:   10000               ;POWER OF 10 TABLE
        1000
        100
        10
        1
        0                   ;END OF TABLE INDICATION
        .RDX 8
POINT:  0                   ;POINTER TO CURRENT POWER OF 10
COUNT:  0                   ;RUNNING SUM WORD
MIN:    "-                  ;ASCII -
POS:    "+                  ;ASCII +
ZERO:   60                  ;ASCII ZERO
TENA:   TENT                ;ADDRES OF POWER OF 10 TABLE
        .END
```

```
;*************************************
;
;        SUBROUTINE PRNT
;
;        AT THE END OF EACH TRIAL PRINTS THE RESULTS
;        INCLUDING TRIAL NUMBER
;                  DISPLAY ADDRESS
;                  CORRECT OR NOT CORRECT
;                  RESPONCE TIME
;                  TEST OR PRACTICE MODE
;
;*************************************
        .TITL   PRNT
        .ENT    .PRNT
        .EXTD   TRYN,DADD,CORTF,TIME,TESTF,.BAD,.WLIN
        .ZREL
.PRNT:  PRNT               ;PAGE ZERO ENTRY
        .NREL
PRNT:   STA     1,AC1      ;STORE AC'S
        STA     2,AC2
        STA     3,AC3
        LDA     1,TRYN     ;GET THE TRIAL NUMBER
        LDA     2,.L1      ;LOAD OUTPUT ARRAY POINTER
        JSR@    .BAD       ;CONVERT TRYN TO ASCII
        LDA@    1,DADD     ;GET FIRST WORD OF DISPLAY
        LDA     2,.L5      ;LOAD OUTPUT ARRAY POINTER
        JSR@    .BAD       ;CONVERT ADDRESS TO ASCII
        LDA     1,COR      ;ASSUME RESPONCE WAS CORRECT
        LDA     2,CORTF    ;GET THE CORRECT FLAG
        MOVR    2,2,SNC    ;SKIP IF IT WAS CORRECT
        LDA     1,NCOR     ;IT WASN'T SO GET "N"
        STA     1,L1+10    ;STORE ASCII N OR C
        LDA     1,TIME     ;GET RESPONCE TIME
        LDA     2,.L10     ;LOAD OUTPUT ARRAY POINTER
        JSR@    .BAD       ;CONVERT TIME TO AXCII
        LDA     1,PRA      ;LOAD A "P"
        LDA     2,TESTF    ;GET THE TEST FLAG
        MOVR    2,2,SZC    ;SKIP IF PRACTICE MODE
        LDA     1,TST      ;TEST,SO LOAD A "T"
        STA     1,L1+15    ;STORE P OR T
;NOW INSERT BLANKS WHERE THEY BELONG
        LDA     1,BL       ;GET CODE FOR A BLANK
        STA     1,L1+3     ;STORE BLANKS IN OUTPUT ARRAY
        STA     1,L1+7
        STA     1,L1+14
        SUB     1,1        ;GENERATE A ZERO
        STA     1,L1+16    ;STORE IN LAST WORD
;OUTPUT LINE IS BUILT, SO PRINT IT
        LDA     2,.L1      ;LOAD OUTPUT POINTER
        JSR@    .WLIN      ;PRINT THE LINE
        LDA     1,AC1      ;RESTORE THE AC'S
        LDA     2,AC2
        JMP@    AC3        ;RETURN
AC1:    0                  ;STORAGE FOR ACCUMULATORS
AC2:    0
AC3:    0
```

```
COR:      020103          ;" C"
NCOR:     020116          ;" N"
TST:      020124          ;" T"
PRA:      020120          ;" P"
BL:       20040           ;ASCII CODE FOR A BLANK,BLANK
.L1:      L1              ;POINTER TO START OF OUTPUT ARRAY
.L5:      L1+4            ;POINTER TO L5
.L10:     L1+11           ;POINTER TO L10
L1:       .BLK    17      ;15 WORDS FOR OUTPUT ARRAY
          .END
```

```
;**************************************************
;
;         VSAF SYSTEM INTERRUPT CONTROL ROUTINES
;                                                       :
;
;         MASTER INTERRUPT CONTROLLER
;         RESPONCE INTERRUPT PROCRSSOR
;         CLOCK INTERRUPT PROCESSOR
;
;**************************************************
          .TITL    INTPR
          .ENT     .INTP,TIME,TIME1,OTIME
          .ENT     .CLON,CRATE,CLONF
          .EXTD    RSTAT,DONEP
          .ZREL                 ;PAGE ZERO ENTRIES
CRATE:    3                     ;SET RTC RATE AT 1000HZ
CLONF:    0                     ;1=RTC IS ON
TIME:     0                     ;COUTNER FOR RTC PULSES
TIME1:    0                     ;USED FOR TIME COMPARISON
OTIME:    500                   ;DISPLAY ON TIME IN MILLI SECS
.CLON:    CLON                  ;POINTER TO CLOCK ON ROUTINE
.CLKIP:   CLKIP                 ;CLOCK INTERRUPT PROCESSOR ENTRY
.RESIP:   RESIP                 ;RESPONCE UNIT INTERRUPT ENTRY
IOMSK:    0                     ;SYSTEM I/O MASK
.INTP:    INTP                  ;POINTER TO MASTER INTERRUPT ENTRY
```

```
;
;
;       MASTER INTERRUPT CONTROLLER
;       DETERMINES THE CAUSE OF THE INTERRUPT AND   :
;       BRANCHES TO THE APPROPRIATE DEVICE CONTROLLER
;
        .NREL
INTP:   SKPDZ   RTC     ;SKIP IF CLOCK NOT DONE
        JMP@    .CLKIP  ;CLOCK DONE PROCESS IT
        SKPDZ   40      ;SKIP IF RESPONCE UNIT NOT DONE
        JMP@    .RESIP  ;PROCESS RES UNIT INTERRUPT
;       AT THIS POINT AN UNEXPECTED DEVICE HAS INTERRUPTED
        INTA    0       ;GET THE DIVICE CODE
        HALT
```

```
;
;
;          RESPONES UNIT INTERRUPT PROCESSOR
;          READS THE STATUS, SETS THE SOFTWARE DONE FLAG
;          AND RETURNS
RESIP:     STA       0,ZSAV    ;SAVE AC0
           LDA       0,0       ;GET THE INTERRUPTED LOC
           STA       0,JMPL    ;STORE RETURN ADDRESS
           MOVL      0,0       ;NOW SAVE THE CARRY
           STA       0,RCARY
           DIA       0,40      ;READ THE DEVICE STATUS
           NIOC      40        ;CLEAR THE INTERRUPT
           MOV#      0,0,SNR   ;CHECK FOR ZERO STATUS
           JMP       DONEI     ;BAD STATUS,SO EXIT
           STA       0,RSTAT   ;STORE IT
           SUBZL     0,0       ;GENERATE A 1
           STA       0,DONEF   ;SET THE DONE FLAG
           SUB       0,0       ;GENERATE A ZERO
           STA       0,CLONF   ;ZERO CLOCK ON FLAG
           JMP       .+3       ;GOOD INTERRUT SO DON'T RESTART
DONEI:     NIOS      40        ;CLEAR THE INTERRUPT AND RESTART
           INTEN               ;REENABLE INTERRUPTS
           LDA       0,RCARY   ;RESTORE CARRY
           MOVR      0,0
           LDA       0,ZSAV    ;RESTORE AC0
           JMP@      JMPL      ;RETURN TO INTERRUPTED LOCATION
ZSAV:      0                   ;STORAGE FOR AC0
RCARY:     0                   ;STORAGE FOR CARRY
JMPL:      0                   ;STORAGE FOR RETURN ADDRESS
```

```
;
;
;        CLOCK INTERRUPT SERVICE
;        INCLUDES CLOCK INTERRUPT PROCESSOR          :
;        AND CLOCK START ENTRIES
;
;
CLKIP:   STA     0,CLACO  ;STORE AC0
         MOVL    0,0      ;GET THE CARRY BIT
         STA     0,CLCRY  ;STORE IT
         LDA     0,0      ;GET RETURN ADDRESS
         STA     0,CLRET  ;STORE IT
         LDA     0,CLONF  ;GET CLOCK ON FLAG
         MOV     0,0 SNR  ;SKIP IF CLOCK IS TO BE ON
         JMP     STOP     ;FLAG IS RESET. CLOCK IS OFF
         ISZ     TIME     ;INCREMENT TIME COUNTER
         MOV     1,1      ;NO OP FOR ISZ
         NIOS    RTC      ;RESTART THE CLOCK
DONE:    LDA     0,CLCRY  ;GET THE CARRY
         MOVR    0,0      ;RESTORE IT
         LDA     0,CLACO  ;RESTORE AC0
         INTEN            ;REENABLE INTERRUPTS
         JMP@    CLRET    ;RETURN
STOP:    NIOC    RTC      ;TURN OFF THE CLOCK
         JMP     DONE     ;EXIT
CLACO:   0                ;STORAGE FOR AC0
CLCRY:   0                ;STORAGE FOR CARRY
CLRET:   0                ;STORAGE FOR RETURN
;        CLOCK START ENTRY
CLON:    STA     0,CLOO   ;STORE AC0
         SUBZL   0,0      ;GENERATE A 1
         STA     0,CLONF  ;SET CLOCK ON FLAG
         SUB     0,0      ;GENERATE A 0
         STA     0,TIME   ;RESET TIME COUNT
         LDA     0,CRATE  ;GET CLOCK RATE
         DOAS    0,RTC    ;START THE CLOCK
         LDA     0,CLOO   ;RESTORE AC0
         JMP     0,3      ;RETURN
CLOO:    0                ;STORAGE FOR AC0
         .END             ;END OF INTERRUPT CONTROL ROUTINES
```

VITA

Douglas D. Carpenter was born on 27 February 1944 in Carroll, Iowa. As an undergraduate he attended Antioch College, Yellow Springs, Ohio, graduating with a Bachelor of Science degree in 1967. He entered the USAF as a Second Lieutenant in 1968. He has served as a Minute-Man Missile Weapon System maintenance officer, and electronic engineer. Prior to entering the Air Force Institute of Technology in 1975, he served as a systems engineer for the Defense Meteorological Satellite Program.

Permanent address: Coon Rapids, Iowa, 50058

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: GE/EE/76D-17
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): THE SCALING PROBLEM IN VISUAL PATTERN RECOGNITION
5. TYPE OF REPORT & PERIOD COVERED: MS Thesis
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Douglas D. Carpenter
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: **Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB OH 45433**
12. REPORT DATE: December, 1976
13. NUMBER OF PAGES:
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17
JERRAL F. GUESS, Captain, USAF
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern Recognition
Human Visual System
Fourier Transform
Visual Perception

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The human pattern recognition system has been modeled as a system utilizing a low-pass, spatially-filtered, Fourier transform to represent input patterns and stored prototypes. This model is used to infer the existence of possible mechanisms underlying the scale invariant aspect of the human pattern recognition system. Two hypotheses are suggested: scaling (size normalization) of input patterns, or the storage of multiple-prototypes based on size. Experiments are carried out to distinguish between the two

(Cont on p 1473B)

BLOCK 20. Abstract continued
(fr p1473A)

utilized

mechanisms. It is concluded that both mechanisms are utilized at different levels within the visual process. A revised model is developed to adequately account for the experimental results.

1473B